UNIT-II

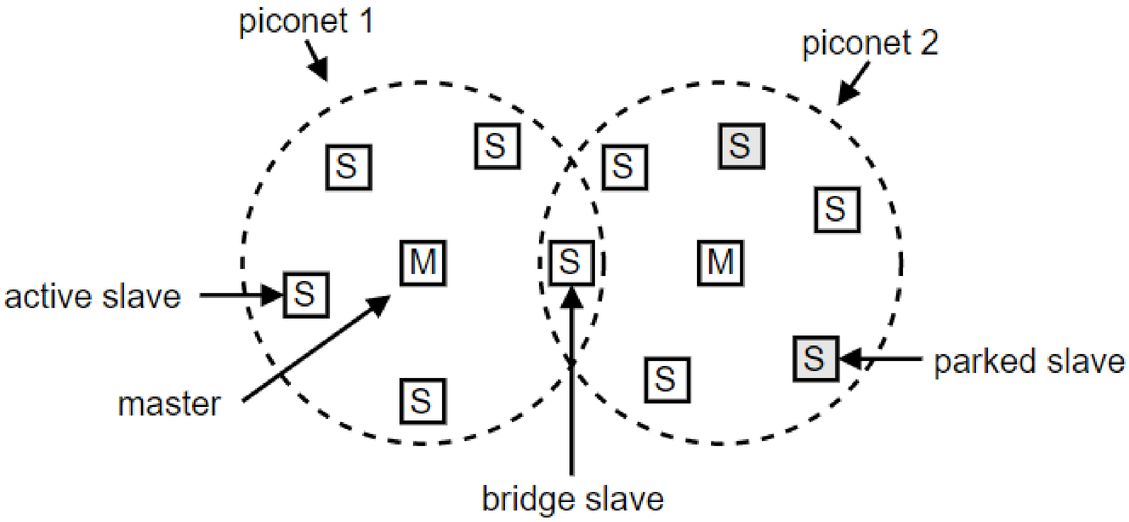
CONTROL UNITS

Bluetooth:

- It is a Wireless Personal Area Network (WPAN) technology and is used for exchanging data over smaller distances.
- This technology was invented by Ericson in 1994.
- It operates in the unlicensed, industrial, scientific and medical (ISM) band from 2.4 GHz to 2.485 GHz.
- Maximum devices that can be connected at the same time are 7. Bluetooth ranges up to 10 meters.
- It provides data rates up to 1 Mbps or 3 Mbps depending upon the version.
- A Bluetooth network is called a **piconet** and a collection of interconnected piconets is called **scatternet**.

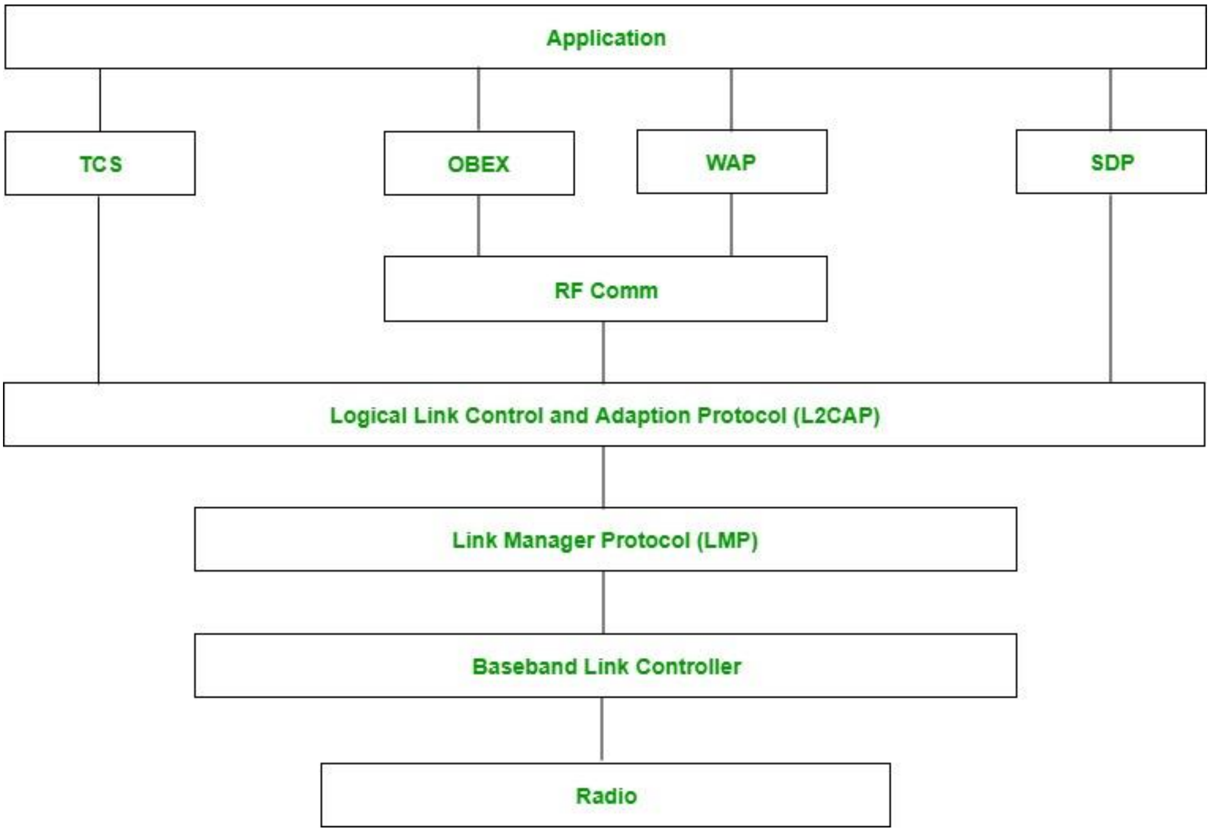
Bluetooth Architecture:

- The basic unit of a Bluetooth system is called a piconet. a piconet consists of a master node, and up to 7 active slave nodes within a distance of 10 meters
- Multiple piconets can exist in the same room, and they can be connected through a bridge node that takes part in multiple piconets. An interconnected collection of piconets is called a scatternet



- There can be up to 255 parked nodes on a piconet. Parked nodes are devices that the master has switched to a low-power state. In parked state a device can't do anything except for respond to an activation or beacon signal from the master
- A piconet is a centralized TDM (Time-Division Multiplexing) system. The master controls the clock and determines which devices get to communicate in which time slot. All communication is between the master and slave, direct slave to slave communication is not possible

Bluetooth protocol stack:


Application
TCS
OBEX
WAP
SDP
RF Comm
Logical Link Control and Adaption Protocol (L2CAP)
Link Manager Protocol (LMP)
Baseband Link Controller
Radio

1. **Radio (RF) layer:** It performs modulation/demodulation of the data into RF signals. It defines the physical characteristics of Bluetooth transceivers. It defines two types of physical links: connection-less and connection-oriented.

2. **Baseband Link layer:** The baseband is the digital engine of a Bluetooth system and is equivalent to the MAC sublayer in LANs. It performs the connection establishment within a piconet.

3. **Link Manager protocol layer:** It performs the management of the already established links which includes authentication and encryption processes. It is responsible for creating the links, monitoring their health, and terminating them gracefully upon command or failure.

4. **Logical Link Control and Adaption Protocol layer:** It is also known as the heart of the Bluetooth protocol stack. It allows the communication between upper and lower layers of the Bluetooth protocol stack. It packages the data packets received from upper layers into the form expected by lower layers. It also performs segmentation and multiplexing.

5. **SDP layer:** It is short for Service Discovery Protocol. It allows discovering the services available on another Bluetooth-enabled device.

6. **RF comm layer:** It is short for Radio Frontend Component. It provides a serial interface with WAP and OBEX. It also provides emulation of serial ports over the logical link control and adaption protocol(L2CAP). The protocol is based on the ETSI standard TS 07.10.

7. **OBEX:** It is short for Object Exchange. It is a communication protocol to exchange objects between 2 devices.

8. **WAP:** It is short for Wireless Access Protocol. It is used for internet access.

9. **TCS:** It is short for Telephony Control Protocol. It provides telephony service. The basic function of this layer is call control (setup & release) and group management for gateway serving multiple devices.

10. **Application layer:** It enables the user to interact with the application.

Advantage:

- Low cost.
- Easy to use.
- It can also penetrate through walls.
- It creates an Adhoc connection immediately without any wires.
- It is used for voice and data transfer.

Disadvantages:

- It can be hacked and hence, less secure.
- It has a slow data transfer rate: of 3 Mbps.
- It has a small range: 10 meters.

Applications:

- Used in laptops, and in wireless PCs.
- In printers.
- In wireless headsets.

ZigBee

- ZigBee is a IEEE 802.15.4 based, low power, low data rate supporting wireless networking standard, which is basically used for two-way communication between sensors and control system.
- It is a short-range communication standard like Bluetooth and Wi-Fi, covering range of 10 to 100 meters.
- The difference being while Bluetooth and Wi-Fi are high data rate communications standard supporting transfer of complex structure like media, software etc.,
- It supports low data rate of about 250 kbps.
- The operating frequencies are 868 MHz, 902 to 928 MHz and 2.4 GHz.
- ZigBee Technology is used mainly for applications requiring low power, low cost, low data rate and long battery life.

Architecture of Zigbee:

The ZigBee Network Protocol follows IEEE 802.15.4 standards for Physical and MAC layers, along with its own Network and Application layers.

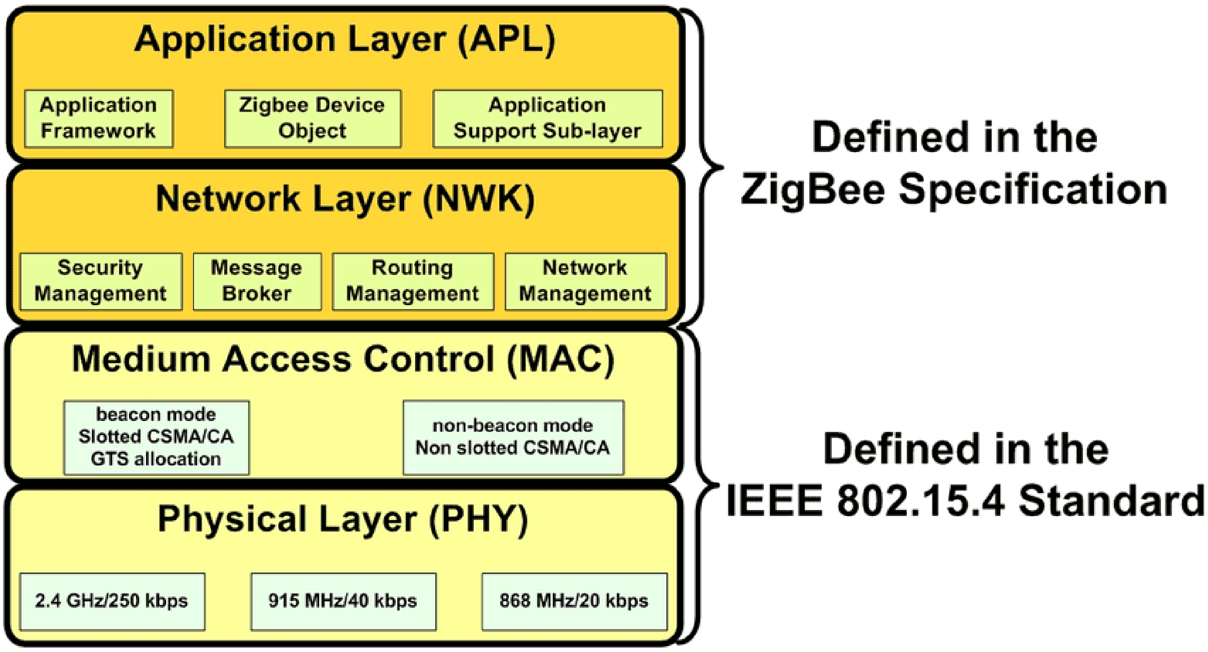


Figure: ZigBee Architecture

Physical Layer:

This layer is responsible for data transmission and reception. Mapping bits of information and permits them to travel through the air by modulation and spreading techniques which is the basic task of physical layer.

Function of Physical Layer in Zigbee Architecture

Physical Layer is responsible for the following functions:

- Activation and deactivation of transmission and reception.
- Channel selection and its assessment.
- Sending and receiving of packets.
- Energy detection within the channel.

Medium Access Control or MAC Layer:

This layer provides interface between the physical and network layers. Data handling and data management are the two main functions of the MAC layer. **Data handling** includes functions such as **"Data Request"** and **"Data Confirm".** The MAC layer adds destination address and transmits options for the outgoing data frames.

When the Zigbee network layer calls the "data request" function, the data gets formatted into relevant MAC header and frame length is added which is the physical header. The data frame is ready to be transmitted.

The purpose of "Data Confirm" function is to communicate the status of the transmitted data. It sends a failure status when the transmission frames exceeds or when there is no response to transmitted data.

Function of Medium Access Control (MAC) Layer in Zigbee Architecture

Medium Access Control (MAC) Layer is responsible for the following:

- Beacon generation and management.
- CSMA-CA (Carrier Sense Multiple Access with Collision Avoidance) is implemented.
- Data frame validation and acknowledgement.
- Data transfer for upper layers.

Network Layer:

Network Layer provides interface between MAC layer and the application layer. It is responsible for routing and establishing different Zigbee network topologies.

When a coordinator attempts to establish a Zigbee network, an energy scan is initiated to find the best RF channel for its new network. When a channel has been chosen, the coordinator assigns a PAN-ID which will be applied to all the devices that join the network.

PAN-ID is a 16-bit number that is used as a network identifier. A node is allowed to communicate on a network only when it undergoes the association process. The association function is used to join a node to a parent.

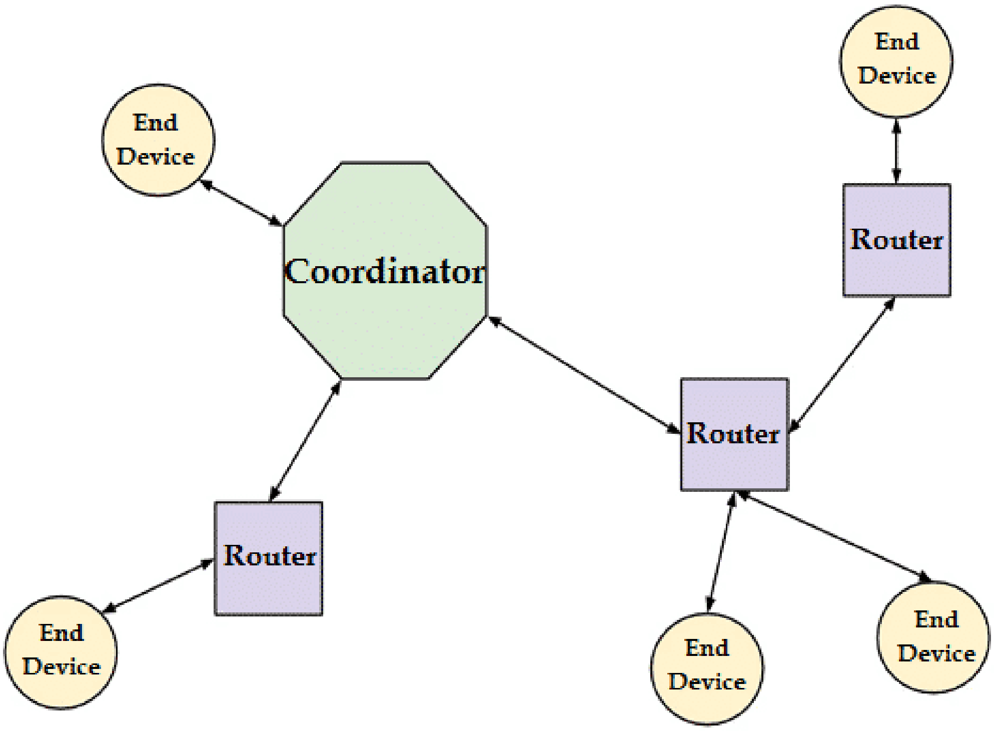


Fig. – Types of Network Nodes in Zigbee Architecture (Zigbee Stack)

When a node loses its parent, it is considered as an orphan device. It usually occurs when the end device is mobile and is out of range or through a failure on the parent. In such case, an orphan scan is performed by broadcasting an “orphan notification” command frame which helps the device to find its parent.

If the parent gets the notification command, it responds back to the device that it exists, and the orphan can rejoin the parent. Thus, the authenticity and confidentiality of a transmission is ensured by the network layer.

Function of Network Layer in Zigbee Architecture

Network Layer in Zigbee architecture is responsible for the following functions:

- Initiation of a network
- Assigning node addresses
- Configuring of new devices
- Providing secured transmission

Application Layer:

The Application Layer in Zigbee architecture consists of sub layers namely:

- Application Support Sub Layer
- Application Framework

Application Support Sub Layer (APS)

This layer is responsible for filtering of packets for end devices, checks for duplicity of packets which is common in a network that supports automatic retries. To maximize the chance of successful transmission, it performs automatic retries, when the acknowledgement is requested by the sender.

It is involved in maintaining binding tables. Binding is the connection between the endpoint on the node to one or more endpoints on other nodes. The address mapping associates a 64-bit MAC address with a Zigbee 16-bit network address.

Function of Application Support Sub Layer (APS)

Application Support Sub Layer (APS) is responsible for the following functions:

- Maintaining binding tables.
- Address definition, mapping, and management.
- Ensuring communication between devices.
- Filtering out packets for non-registered end devices or profiles that don’t match and reassembling of the packets.

Application Framework

The Application Framework depends on the vendor who has chosen for specific applications to interact with Zigbee protocol. This represents how end points are implemented, how data requests and data confirmation is executed for that particular vendor.

Zigbee Applications:

1. Home Automation
2. Medical Data Collection
3. Industrial Control Systems
4. meter reading system
5. light control system

Wi-Fi

- WiFi stands for Wireless Fidelity.
- It is based on the IEEE 802.11 family of standards and is primarily a local area networking (LAN) technology designed to provide in-building broadband coverage.

- Current WiFi systems support a peak physical-layer data rate of 54 Mbps and typically provide indoor coverage over a distance of 100 feet.

IEEE 802.11 Architecture

The components of an IEEE 802.11 architecture are as follows

1) Stations (STA) − Stations comprise all devices and equipments that are connected to the wireless LAN. A station can be of two types:

- **Wireless Access Points (WAP)** − WAPs or simply access points (AP) are generally wireless routers that form the base stations or access.
- **Client.** − Clients are workstations, computers, laptops, printers, smartphones, etc.

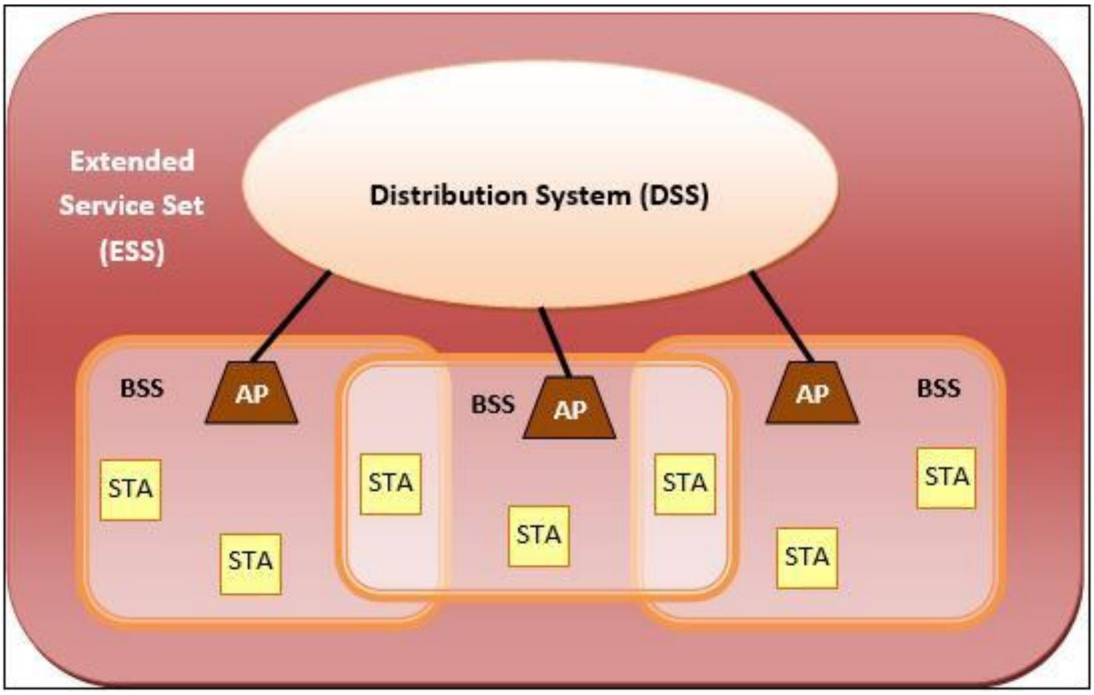
Each station has a wireless network interface controller.

2) Basic Service Set (BSS) −A basic service set is a group of stations communicating at physical layer level. BSS can be of two categories depending upon mode of operation:

- **Infrastructure BSS** − Here, the devices communicate with other devices through access points.
- **Independent BSS** − Here, the devices communicate in peer-to-peer basis in an ad hoc manner.

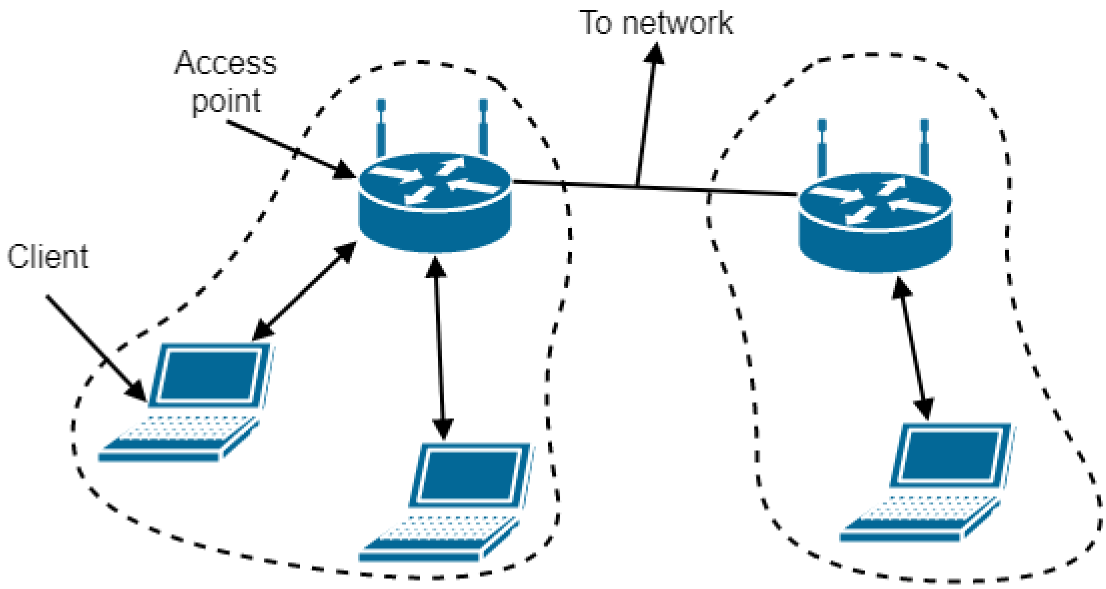
3) Extended Service Set (ESS) − It is a set of all connected BSS.

4) Distribution System (DSS) − It connects access points in ESS.

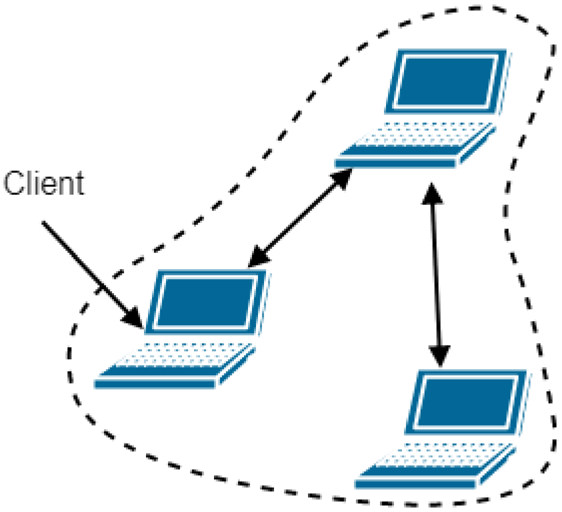


802.11 networks can be used in two modes: infrastructure mode and ad-hoc mode

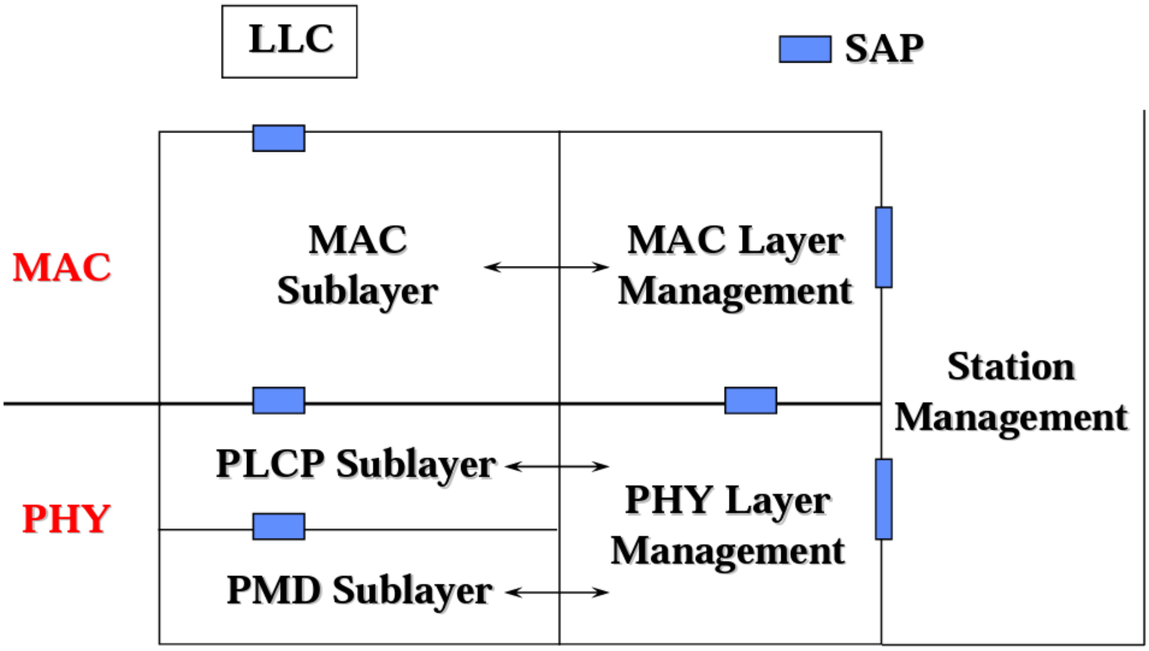
Infrastructure mode uses an AP (Access Point) that is connected to the network. Clients send and receive packets via the AP. Several APs can be connected to form an extended network



Ad-hoc mode is a collection of computers connected to each other so that they can send frames to each other. There's no AP



802.11 Protocol Entities



- **MAC Entity**

– basic access mechanism

– fragmentation/defragmentation

– encryption/decryption

- **MAC Layer Management Entity**

– synchronization– power management

– roaming– MAC MIB

- **Physical Layer Convergence Protocol (PLCP)**

– PHY-specific, supports common PHY SAP

– provides Clear Channel Assessment signal (carrier sense)

- **Physical Medium Dependent Sublayer (PMD)**

– modulation and encoding

- **PHY Layer Management**

– channel tuning (channel switching delay: 224us in 802.11b)

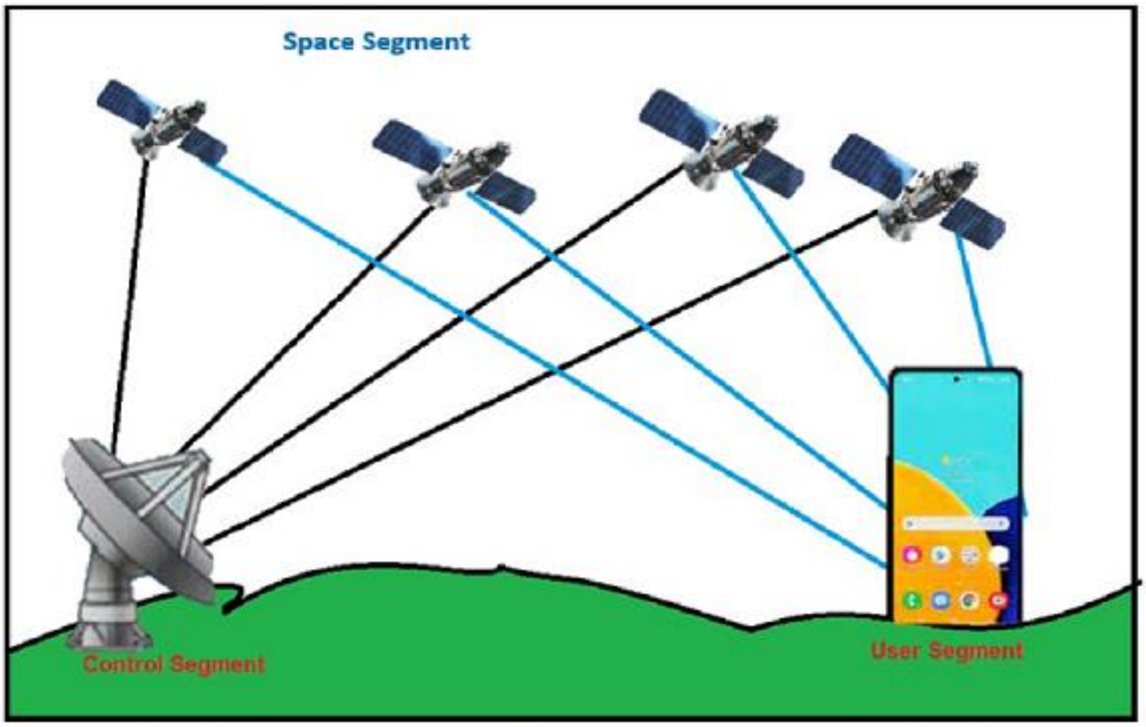
– PHY MIB

- **Station Management**

– interacts with both MAC Management and PHY Management

GPS

- **GPS** stands for **Global Positioning System.** GPS is a satellite-based navigation system that allows ground users to provide their exact location, velocity, and time 24 hours a day, in all weather conditions, all over the world.
- GPS developed by the U.S. Department of Défense. It was basically designed to assist soldiers and military
- The GPS systems are a Combination of a network of satellites that are constantly sending coded information in the form of radio signals. After that receiver received the signals and interprets the transmitted information from the satellites to locate the position on earth accurately.



Global Positioning System

Architecture of GPS:

GPS Architecture is basically divided into three segments.

- Space Segment (SS)
- Control Segment (CS)
- User Segment (US)

Space segment:

- The space segment (SS) is composed of the orbiting GPS satellites, or Space Vehicles (SV).
- The GPS design originally called for 24 SVs, eight each in three circular orbital planes.
- This was modified to six planes with four satellites each.

- The orbital planes are centered on the Earth.
- The six planes have approximately 55° inclination and are separated by 60° right ascension of the ascending node.
- The orbits are arranged so that at least six satellites are always within line of sight from almost everywhere on Earth's surface.
- Satellites are orbiting at an altitude of approximately 20,200 kilometers.
- Each SV makes two complete orbits each sidereal day, repeating the same ground track.
- As of March 2008, there are 31 actively broadcasting satellites in the GPS constellation.
- The additional satellites improve the precision of GPS receiver calculations by providing redundant measurements.

Control Segment:

- The GPS-System is controlled by the US Army. The “master control station” and four additional monitoring stations were set up for monitoring the satellites.
- The Control Segment is composed of
    1. A Master Control Station (MCS)
    2. An Alternate Master Control Station,
    3. Four dedicated Ground Antennas
    4. Six dedicated Monitor Stations
- The flight paths of the satellites are tracked by dedicated U.S. Air Force monitoring stations in
    1. Hawaii
    2. Kwajalein
    3. Ascension Island
    4. Diego Garcia
    5. Colorado Springs
- Monitor stations operated in England, Argentina, Ecuador, Bahrain, Australia and Washington DC.
- During August and September 2005, six more monitor stations of the National Geospatial-Intelligence Agency were added to the grid. Now, every satellite can be seen from at least two monitor stations.
- This allows calculating more precise orbits and ephemeris data. For the end user, a better position precision can be expected from this.
- In the near future, five more stations will be added so that every satellite can be seen by at least three monitor stations. This improves integrity monitoring of the satellites and thus the whole system.

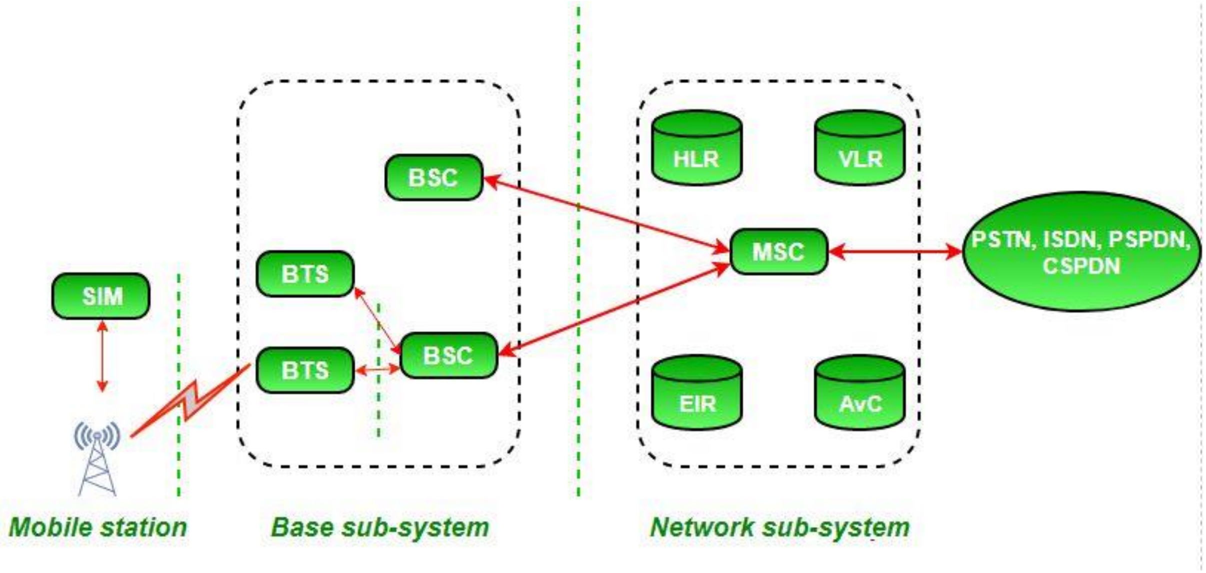
User Segment:

- The User Segment is composed of hundreds of thousands of U.S. and allied military users of the secure GPS Precise Positioning Service, and tens of millions of civil, commercial and scientific users of the Standard Positioning Service.
- In general, GPS receivers are composed of an antenna, tuned to the frequencies transmitted by the satellites, receiver-processors, and a highly-stable clock.

GSM

- GSM stands for Global System for Mobile communication. Today, GSM is used by more than 800 million end-users spread across 190 countries which represent around 70 percent of today's digital wireless market. So, let's see how it works.
- In GSM, the geographical area is divided into hexagonal cells whose side depends upon the power of the transmitter and load on the transmitter (number of end-user). At the center of the cell, there is a base station consisting of a transceiver (combination of transmitter and receiver) and an antenna.

Architecture :



Function of Components :

1. **Mobile station (MS) :** It refers for mobile station. Simply, it means a mobile phone.
2. **Base transceiver system (BTS) :** It maintains the radio component with MS.
3. **Base station controller (BSC) :** Its function is to allocate necessary time slots between the BTS and MSC.
4. **Home location register (HLR) :** It is the reference database for subscriber parameters like subscriber's ID, location, authentication key, etc.
5. **Visitor location register (VLR) :** It contains a copy of most of the data stored in HLR which is temporary and exists only until the subscriber is active.
6. **Equipment identity register (EIR) :** It is a database that contains a list of valid mobile equipment on the network.
7. **Authentication center (AuC) :** It performs authentication of subscribers.

Working :

GSM is combination of TDMA (Time Division Multiple Access), FDMA (Frequency Division

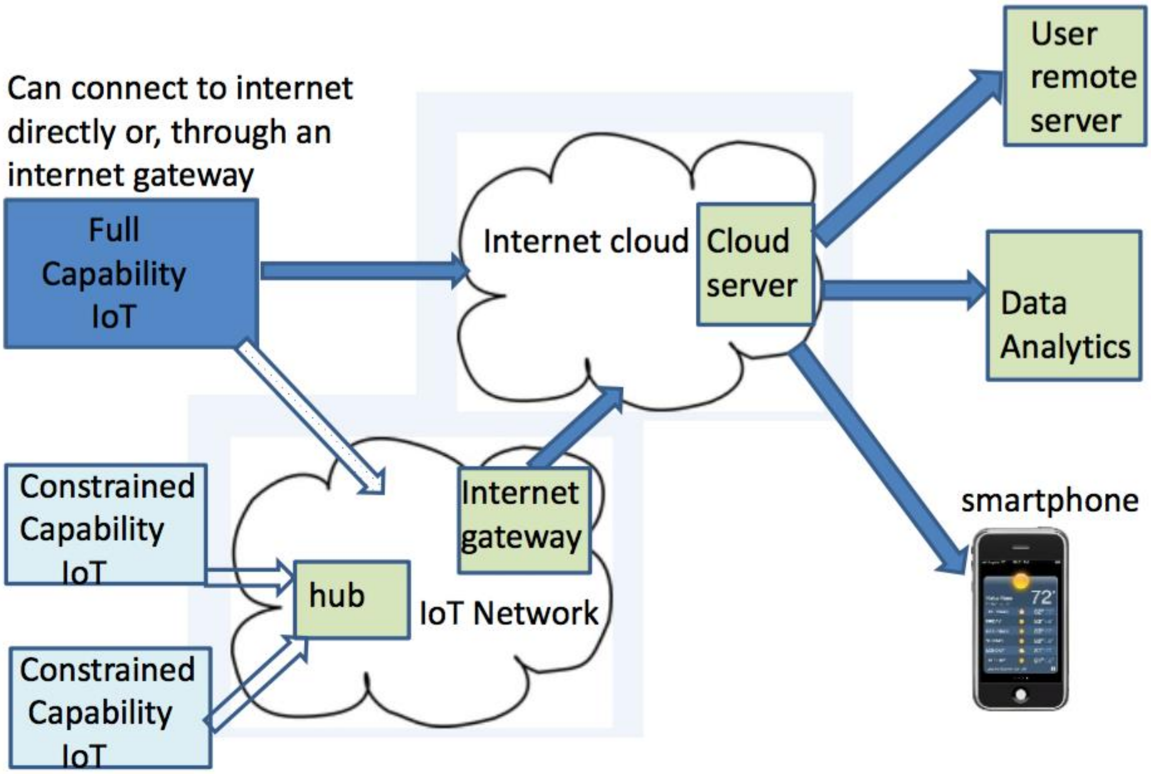
Multiple Access) and Frequency hopping. Initially, GSM use two frequency bands of 25 MHz width : 890 to 915 MHz frequency band for up-link and 935 to 960 MHz frequency for down-link. Later on, two 75 MHz band were added. 1710 to 1785 MHz for up-link and 1805 to 1880 MHz for down-link. up-link is the link from ground station to a satellite and down-link is the link from a satellite down to one or more ground stations or receivers. GSM divides the 25 MHz band into 124 channels each having 200 KHz width and remaining 200 KHz is left unused as a guard band to avoid interference.

Control channels : These are main control channels in GSM :

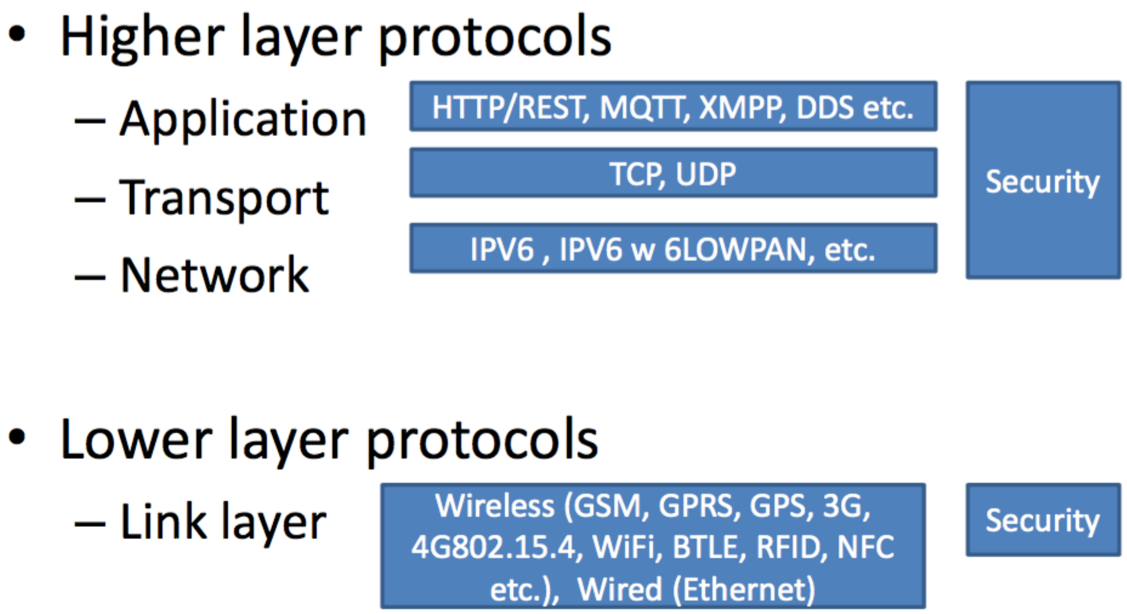
1. **BCH** (Broadcast Channel) : It is for down-link only. It has following types –
   1. **BCCH** (Broadcast Control Channel) : It broadcasts information about the serving cell.
   2. **SCH** (Synchronization channel) : Carries information like frame number and BSIC (Base Station Identity Code) for frame synchronization.
   3. **FCCH** (Frequency Correction Channel) : Enable MS to synchronize to frequency.
2. **CCCH** (Common Control Channel) : It has following types –
   1. **RACH** (Random Access Channel) : Used by MS when making its first access to network. It is for up-link only.
   2. **AGCH** (Access Grant Channel) : Used for acknowledgement of the access attempt sent on RACH. It is for down-link only.
   3. **PCH** (Paging Channel) : Network page the MS, if there is an incoming call or a short message. It is for down-link only.
3. **DCCH** (Dedicated Control Channel) : It is for both up-link and down-link. It has following types –
   1. **SDCCH** (Stand-alone Dedicated Control Channel) : It is used for call setup, authentication, ciphering location update and SMS.
   2. **SACCH** (Slow Associated Control Channel) : Used to transfer signal while MS have ongoing conversation on topic or while SDCCH is being used.
   3. **FACCH** (Fast Associated Control Channel) : It is used to send fast message like hand over message.

IoT Protocol Stack

Simplified IoT System Architecture

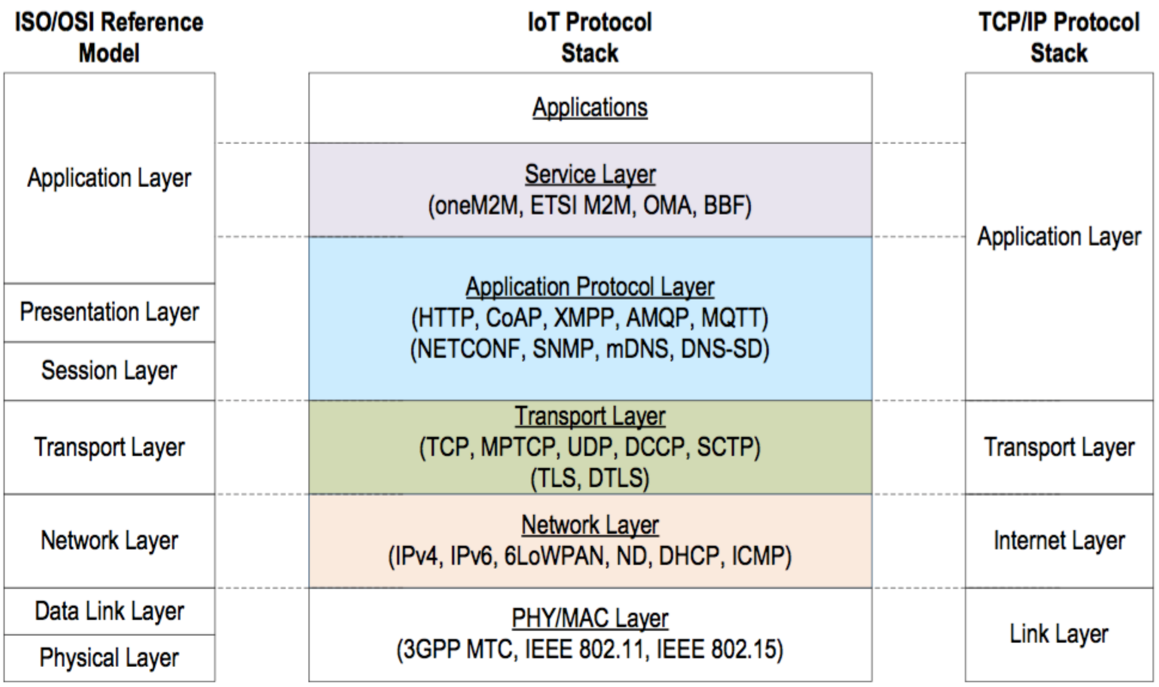


IoT Levels



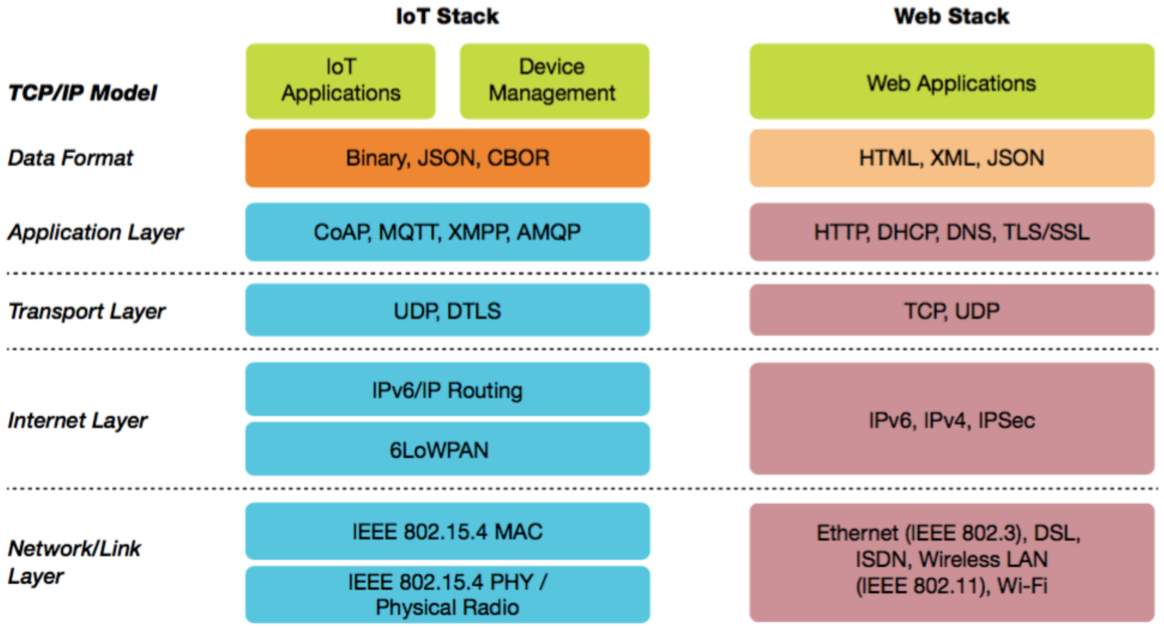
- Higher layer protocols
  - Application — HTTP/REST, MQTT, XMPP, DDS etc.
  - Transport — TCP, UDP
  - Network — IPV6 , IPV6 w 6LOWPAN, etc.
  - Security
- Lower layer protocols
  - Link layer — Wireless (GSM, GPRS, GPS, 3G, 4G802.15.4, WiFi, BTLE, RFID, NFC etc.), Wired (Ethernet)
  - Security

IoT Protocols and Standards



ISO/OSI Reference Model	IoT Protocol Stack	TCP/IP Protocol Stack
Application Layer	Applications	Application Layer
	Service Layer (oneM2M, ETSI M2M, OMA, BBF)	
Presentation Layer Session Layer	Application Protocol Layer (HTTP, CoAP, XMPP, AMQP, MQTT) (NETCONF, SNMP, mDNS, DNS-SD)	
Transport Layer	Transport Layer (TCP, MPTCP, UDP, DCCP, SCTP) (TLS, DTLS)	Transport Layer
Network Layer	Network Layer (IPv4, IPv6, 6LoWPAN, ND, DHCP, ICMP)	Internet Layer
Data Link Layer Physical Layer	PHY/MAC Layer (3GPP MTC, IEEE 802.11, IEEE 802.15)	Link Layer

Standardized IoT Protocols



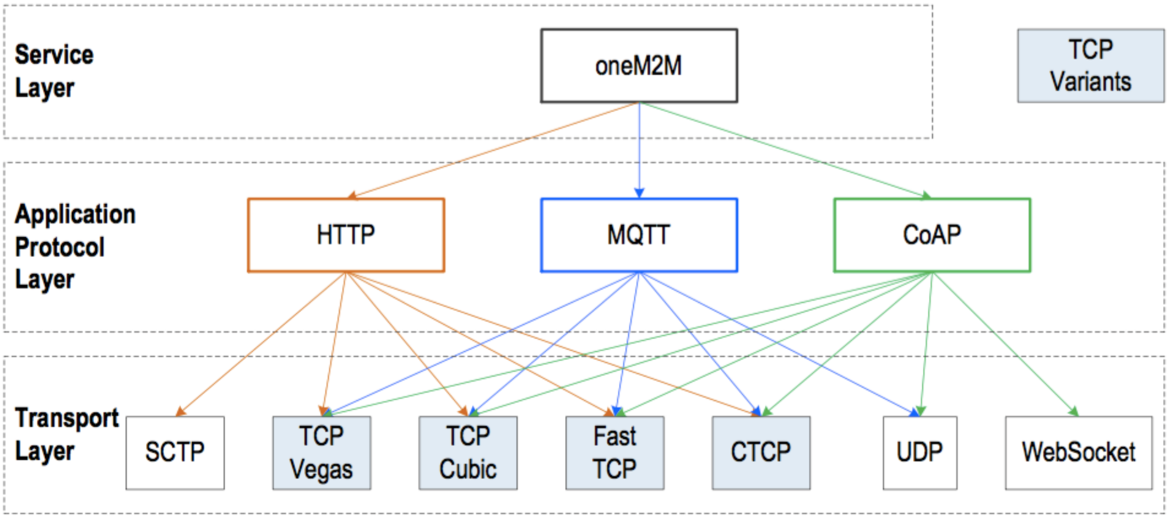
	IoT Stack		Web Stack
TCP/IP Model	IoT Applications	Device Management	Web Applications
Data Format	Binary, JSON, CBOR		HTML, XML, JSON
Application Layer	CoAP, MQTT, XMPP, AMQP		HTTP, DHCP, DNS, TLS/SSL
Transport Layer	UDP, DTLS		TCP, UDP
Internet Layer	IPv6/IP Routing 6LoWPAN		IPv6, IPv4, IPSec
Network/Link Layer	IEEE 802.15.4 MAC IEEE 802.15.4 PHY / Physical Radio		Ethernet (IEEE 802.3), DSL, ISDN, Wireless LAN (IEEE 802.11), Wi-Fi

IoT Protocols

- **Infrastructure** (ex: 6LowPAN, IPv4/IPv6, RPL)
- **Identification** (ex: EPC, uCode, IPv6, URIs)
- **Comms / Transport** (ex: Wifi, Bluetooth, LPWAN)
- **Discovery** (ex: Physical Web, mDNS, DNS-SD)
- **Data Protocols** (ex: MQTT, CoAP, AMQP, Websocket, Node)

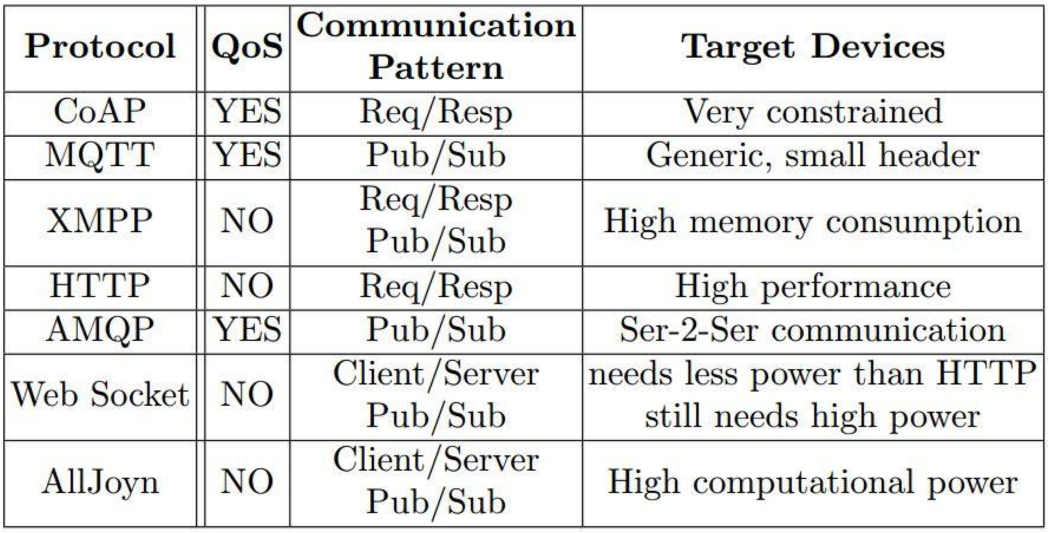
- **Device Management** (ex: TR-069, OMA-DM)
- **Semantic** (ex: JSON-LD, Web Thing Model)
- **Multi-layer Frameworks** (ex: Alljoyn, IoTivity, Weave, Homekit)

IoT Protocol Stack



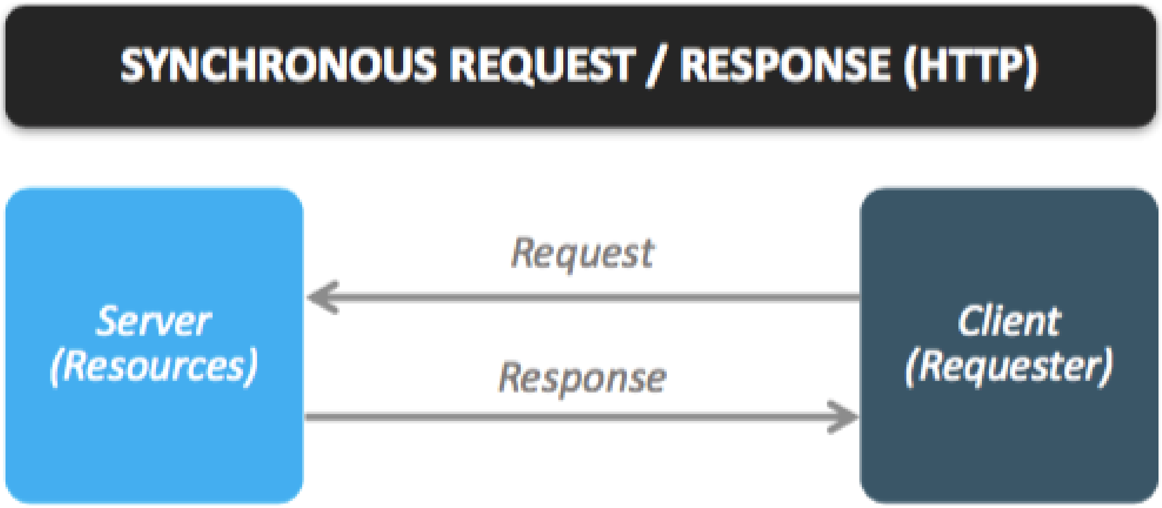
IoT Application Layer Protocols

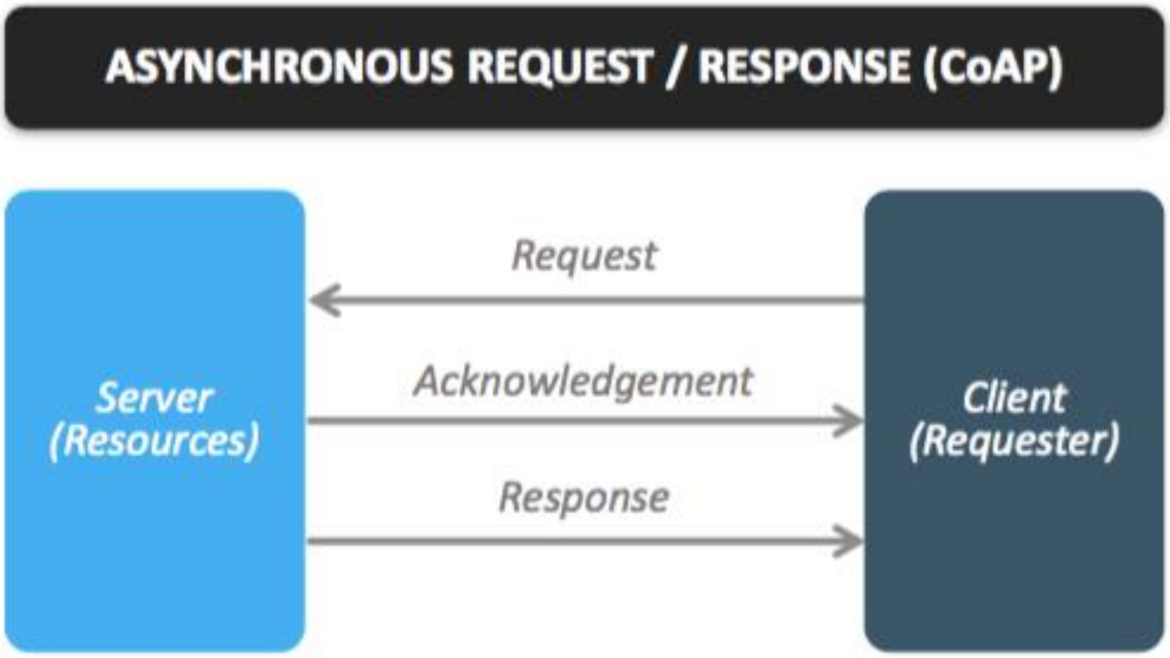
- Most popular application layer protocols used nowadays:
  - **CoAP:** Constrained Application Protocol
  - **MQTT:** Message Queuing Telemetry Transport
  - **XMPP:** Extensible Messaging and Presence Protocol
  - **AMQP:** Advanced Message Queuing Protocol
  - **WebSocket:** Computer Communications Protocol
  - **Alljoyn:** Full stack of protocols intended for IoT. Not separable application layer protocol

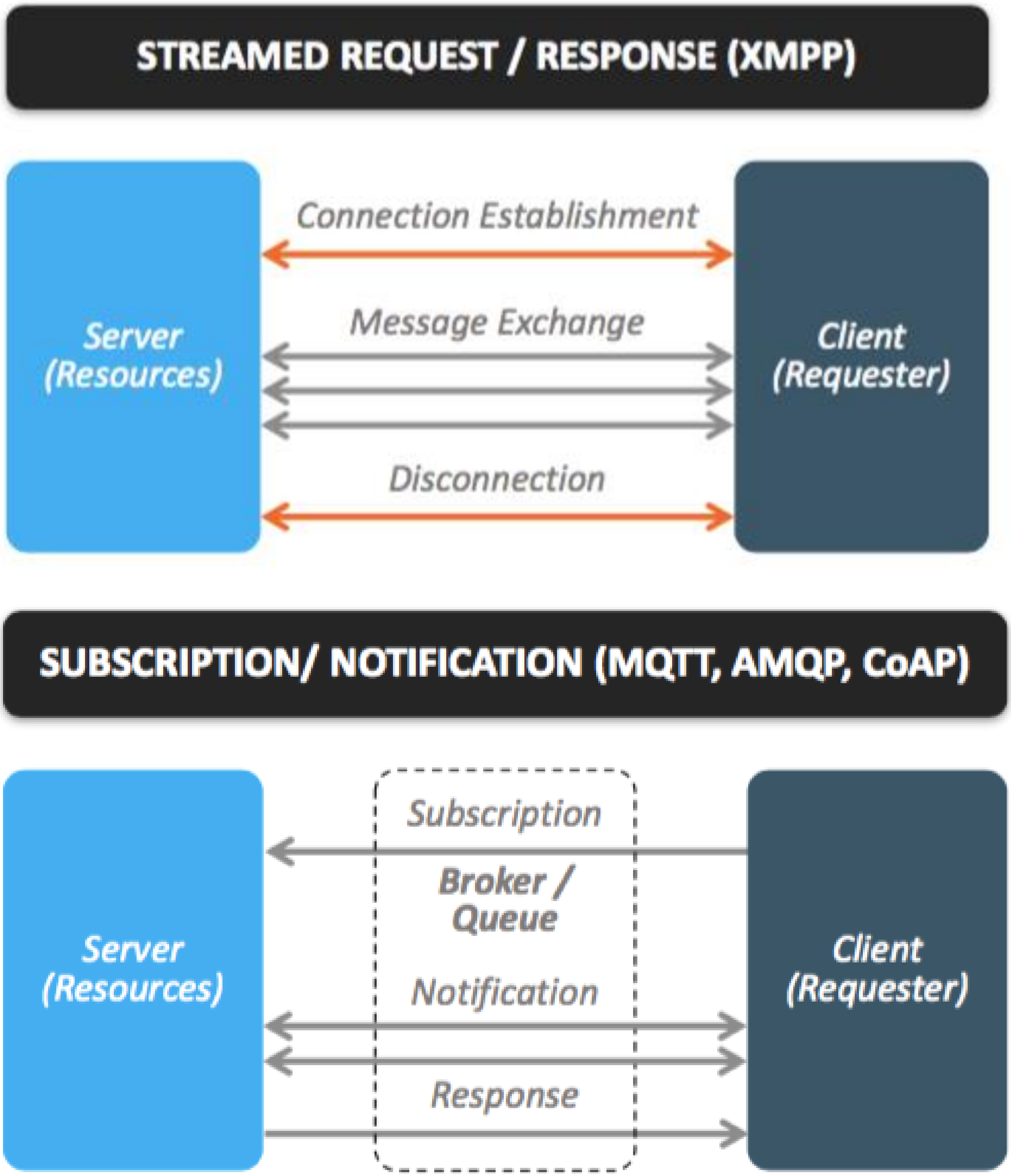


Protocol	QoS	Communication Pattern	Target Devices
CoAP	YES	Req/Resp	Very constrained
MQTT	YES	Pub/Sub	Generic, small header
XMPP	NO	Req/Resp Pub/Sub	High memory consumption
HTTP	NO	Req/Resp	High performance
AMQP	YES	Pub/Sub	Ser-2-Ser communication
Web Socket	NO	Client/Server Pub/Sub	needs less power than HTTP still needs high power
AllJoyn	NO	Client/Server Pub/Sub	High computational power

IoT Messaging Protocols







MQTT - Message Queueing Telemetry Transport

- MQTT stands for **Message Queuing Telemetry Transport**.
- MQTT is a machine-to-machine internet of things connectivity protocol.

- It is an extremely lightweight and publish-subscribe messaging transport protocol.
- It is a publish and subscribe system where we can publish and receive the messages as a client. It makes it easy for communication between multiple devices. It is a simple messaging protocol designed for the constrained devices and with low bandwidth, so it's a perfect solution for the internet of things applications.

MQTT Architecture

To understand the MQTT architecture, we first look at the components of the MQTT.

- **Message**
- **Client**
- **Server or Broker**
- **TOPIC**

Message

The message is the data that is carried out by the protocol across the network for the application. When the message is transmitted over the network, then the message contains the following parameters:

1. Payload data
2. Quality of Service (QoS)
3. Collection of Properties
4. Topic Name

Client

In MQTT, the subscriber and publisher are the two roles of a client. The clients subscribe to the topics to publish and receive messages. In simple words, we can say that if any program or device uses an MQTT, then that device is referred to as a client. A device is a client if it opens the network connection to the server, publishes messages that other clients want to see, subscribes to the messages that it is interested in receiving, unsubscribes to the messages that it is not interested in receiving, and closes the network connection to the server.

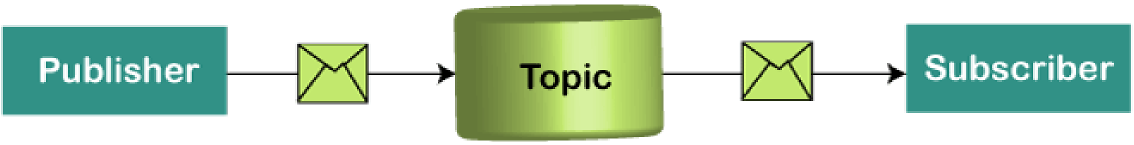
In MQTT, the client performs two operations:

1. **Publish:** When the client sends the data to the server, then we call this operation as a publish.
2. **Subscribe:** When the client receives the data from the server, then we call this operation a subscription.

Server

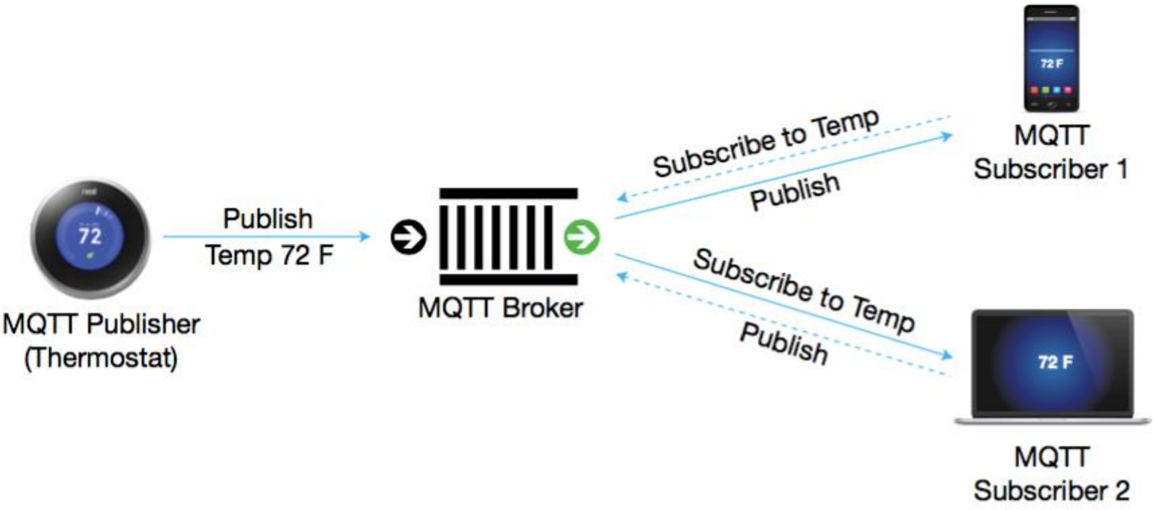
The device or a program that allows the client to publish the messages and subscribe to the messages. A server accepts the network connection from the client, accepts the messages from the client, processes the subscribe and unsubscribe requests, forwards the application messages to the client, and closes the network connection from the client.

TOPIC

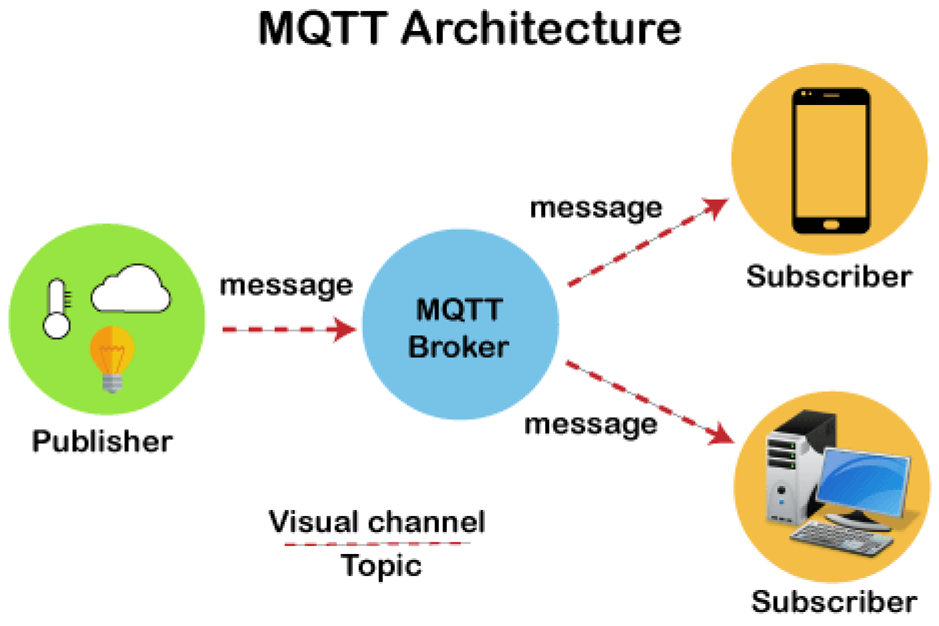


The label provided to the message is checked against the subscription known by the server is known as TOPIC.

MQTT in the world of IoT

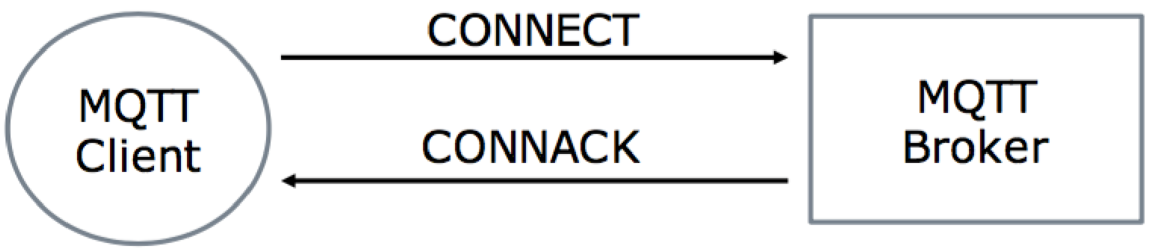


Architecture of MQTT

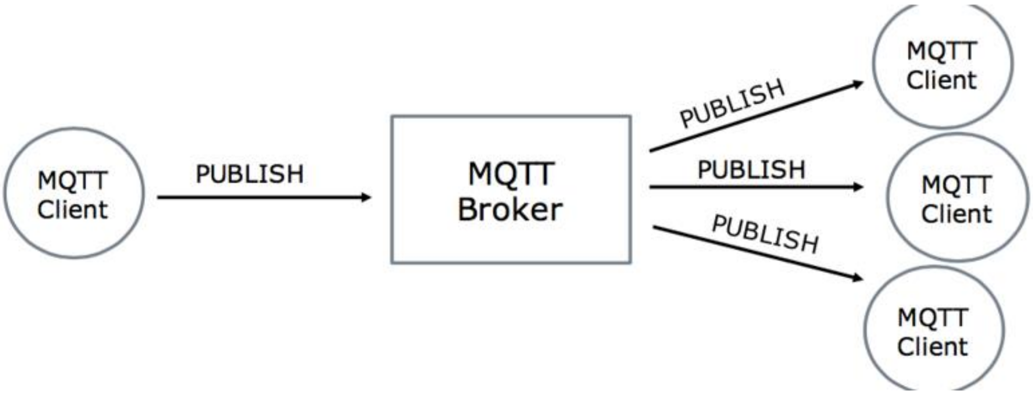


To understand the working of MQTT more clearly, we will look at the example. Suppose a device has a temperature sensor and wants to send the rating to the server or the broker. If the phone or desktop application wishes to receive this temperature value on the other side, then there will be two things that happened. The publisher first defines the topic; for example, the temperature then publishes the message, i.e., the temperature's value. After publishing the message, the phone or the desktop application on the other side will subscribe to the topic, i.e., temperature and then receive the published message, i.e., the value of the temperature. The server or the broker's role is to deliver the published message to the phone or the desktop application.

MQTT Open Connection



MQTT Publishing



Quality of Service (QoS) in MQTT

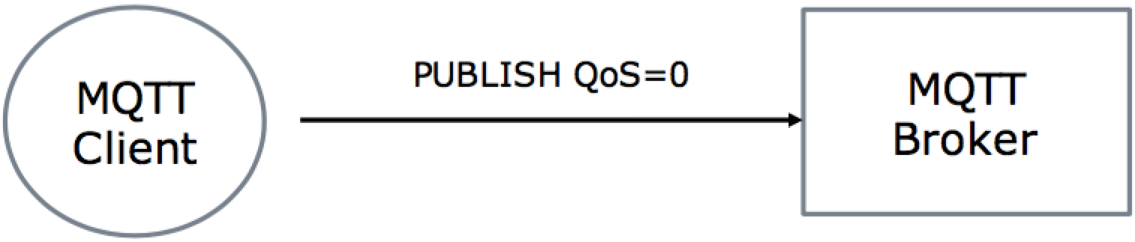
- The Quality of Service (QoS) level is an agreement between the sender of a message and the receiver of a message that defines the guarantee of delivery for a specific message. There are 3 QoS levels in MQTT:
- At most once (0)
- At least once (1)
- Exactly once (2).

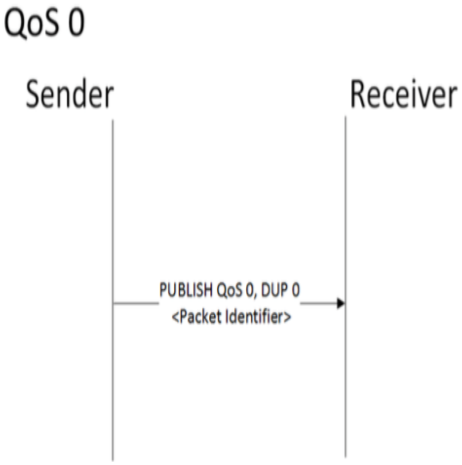
When you talk about QoS in MQTT, you need to consider the two sides of message delivery:

- Message delivery form the publishing client to the broker.
- Message delivery from the broker to the subscribing client.

The client that publishes the message to the broker defines the QoS level of the message when it sends the message to the broker. The broker transmits this message to subscribing clients using the QoS level that each subscribing client defines during the subscription process. If the subscribing client defines a lower QoS than the publishing client, the broker transmits the message with the lower quality of service.

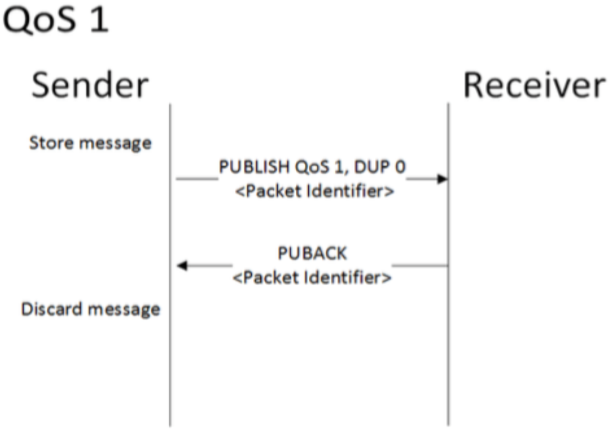
MQTT QoS 0: “at most once”



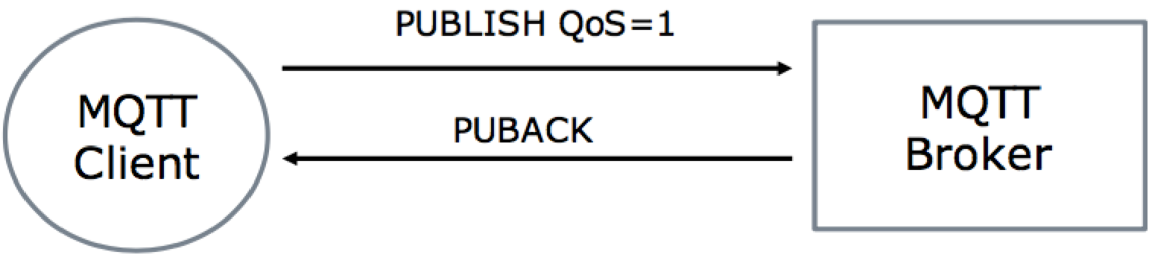


The minimal QoS level is zero. This service level guarantees a best-effort delivery. There is no guarantee of delivery. The recipient does not acknowledge receipt of the message and the message is not stored and re-transmitted by the sender. QoS level 0 is often called "fire and forget" and provides the same guarantee as the underlying TCP protocol.

MQTT QoS 1: "at least once"

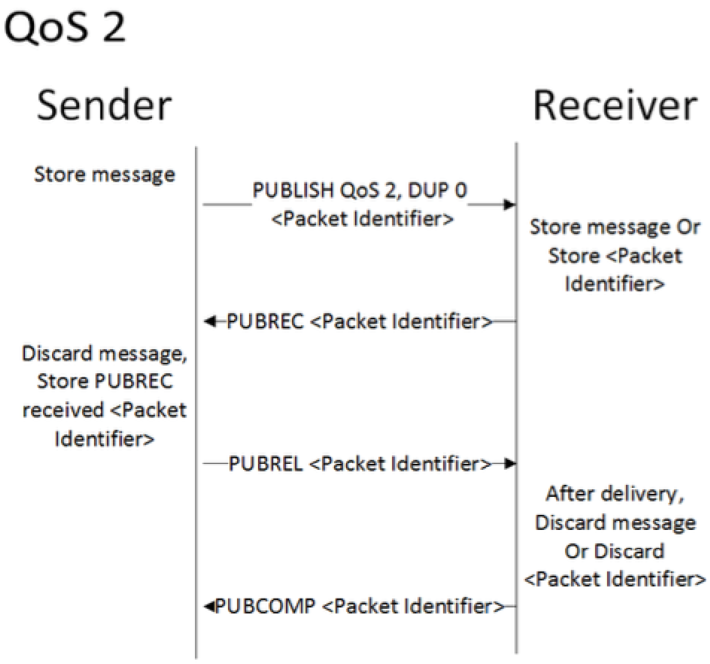


QoS level 1 guarantees that a message is delivered at least one time to the receiver. The sender stores the message until it gets a PUBACK packet from the receiver that acknowledges receipt of the message. It is possible for a message to be sent or delivered multiple times.

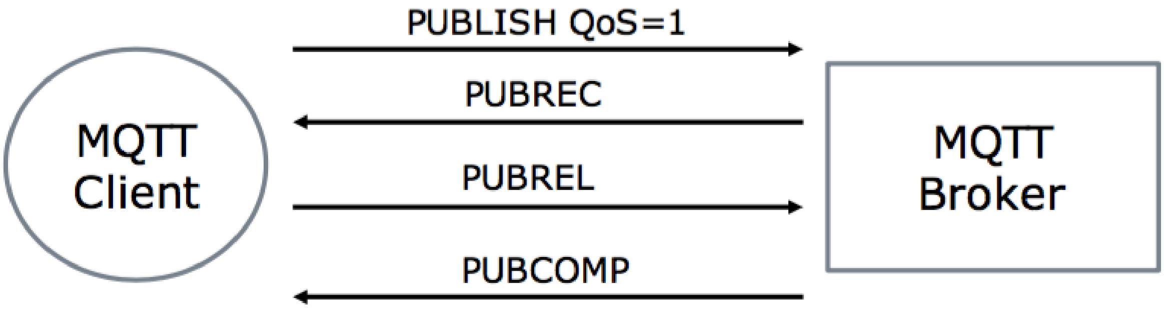


- The sender uses the packet identifier in each packet to match the PUBLISH packet to the corresponding PUBACK packet. If the sender does not receive a PUBACK packet in a reasonable amount of time, the sender resends the PUBLISH packet. When a receiver gets a message with QoS 1, it can process it immediately. For example, if the receiver is a broker, the broker sends the message to all subscribing clients and then replies with a PUBACK packet.

MQTT QoS 2: “exactly once”


QoS 2
Sender
Receiver
Store message
PUBLISH QoS 2, DUP 0 <Packet Identifier>
Store message Or Store <Packet Identifier>
PUBREC <Packet Identifier>
Discard message, Store PUBREC received <Packet Identifier>
PUBREL <Packet Identifier>
After delivery, Discard message Or Discard <Packet Identifier>
PUBCOMP <Packet Identifier>

QoS 2 is the highest level of service in MQTT. This level guarantees that each message is received only once by the intended recipients. QoS 2 is the safest and slowest quality of service level. The guarantee is provided by at least two request/response flows (a four-part handshake) between the sender and the receiver. The sender and receiver use the packet identifier of the original PUBLISH message to coordinate delivery of the message.



- When a receiver gets a QoS 2 PUBLISH packet from a sender, it processes the publish message accordingly and replies to the sender with a PUBREC packet that acknowledges the PUBLISH packet. If the sender does not get a PUBREC packet from the receiver, it sends the PUBLISH packet again with a duplicate (DUP) flag until it receives an acknowledgement

- When a receiver gets a QoS 2 PUBLISH packet from a sender, it processes the publish message accordingly and replies to the sender with a PUBREC packet that acknowledges the PUBLISH packet. If the sender does not get a PUBREC packet from the receiver, it sends the PUBLISH packet again with a duplicate (DUP) flag until it receives an acknowledgement
- When a receiver gets a QoS 2 PUBLISH packet from a sender, it processes the publish message accordingly and replies to the sender with a PUBREC packet that acknowledges the PUBLISH packet. If the sender does not get a PUBREC packet from the receiver, it sends the PUBLISH packet again with a duplicate (DUP) flag until it receives an acknowledgement

CoAP - Constrained Application Protocol

- CoAP stands for Constrained Application Protocol, and it is defined in RFC 7252.
- CoAP is a simple protocol with low overhead specifically designed for constrained devices (such as microcontrollers) and constrained networks.
- This protocol is used in M2M data exchange and is very similar to HTTP.

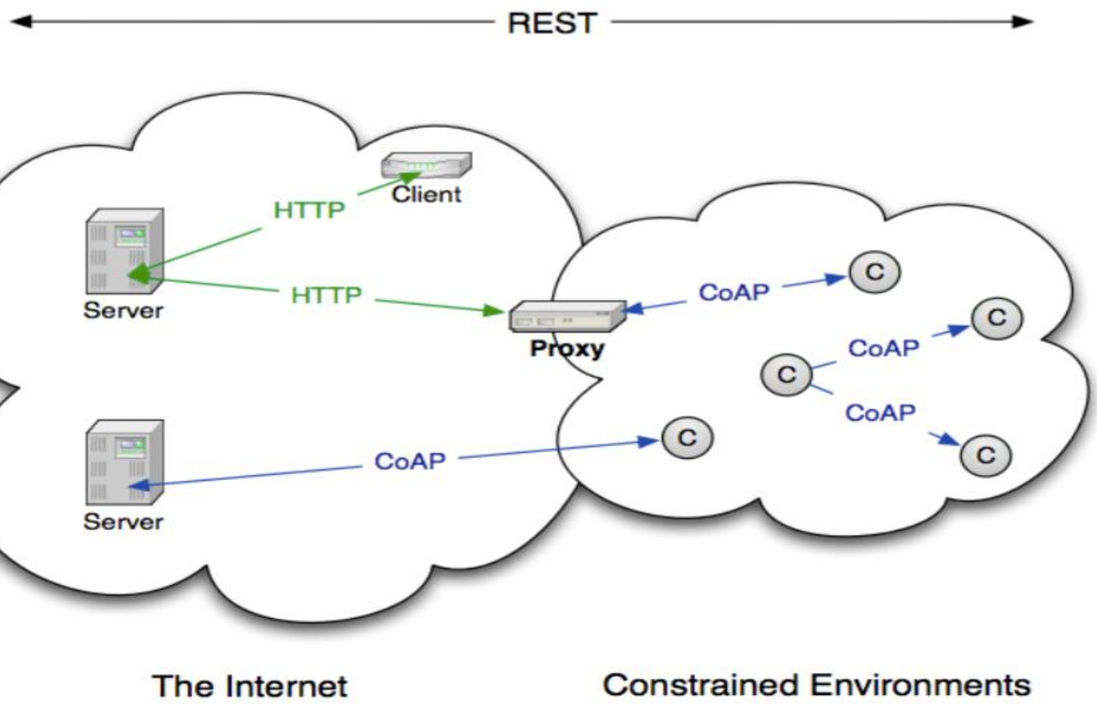
The main features of CoAP protocols are:

- Web protocol used in M2M with constrained requirement
- Asynchronous message exchange
- Low overhead and very simple to parse
- URI and content-type support

- Proxy and caching capabilities

Some features are very similar to HTTP even if CoAP must not be considered a compressed HTTP protocol because CoAP is specifically designed for IoT and in more details for M2M so it is very optimized for this task.

CoAP Architecture



From the abstraction protocol layer, CoAP can be represented as:

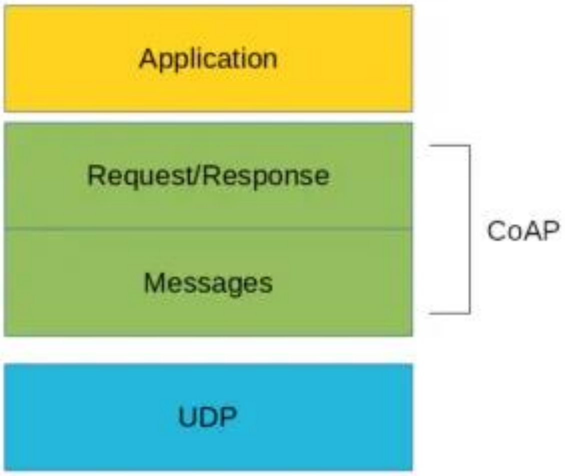


Fig: coap protocol stack

There are two different layers that make CoAp protocol: Messages and Request/Response. The Messages layer deals with UDP and with asynchronous messages. The Request/Response layer manages request/response interaction based on request/response messages.

CoAP supports four different message types:

- Confirmable
- Non-confirmable
- Acknowledgment
- Reset

Before going deeper into the CoAp protocol, structure is useful to define some terms that we will use later:

Endpoint: An entity that participates in the CoAP protocol. Usually, an Endpoint is identified with a host

Sender: The entity that sends a message

Recipient: The destination of a message

Client: The entity that sends a request and the destination of the response

Server: The entity that receives a request from a client and sends back a response to the client

CoAP Messages Model

This is the lowest layer of CoAP. This layer deals with UDP exchanging messages between endpoints. Each CoAP message has a unique ID; this is useful to detect message duplicates. A CoAP message is built by these parts:

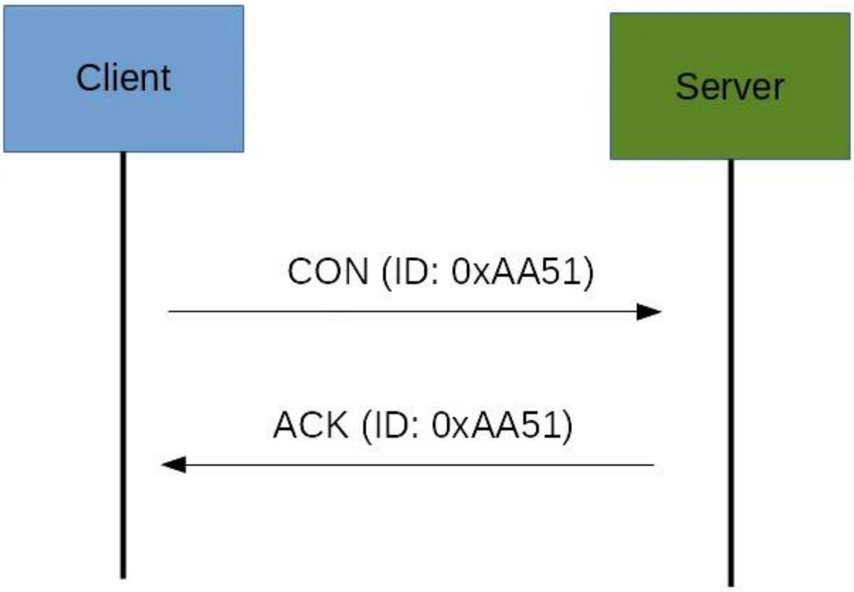
- A binary header
- A compact option
- Payload

As said before, the CoAP protocol uses two kinds of messages:

- Confirmable message
- Non-confirmable message

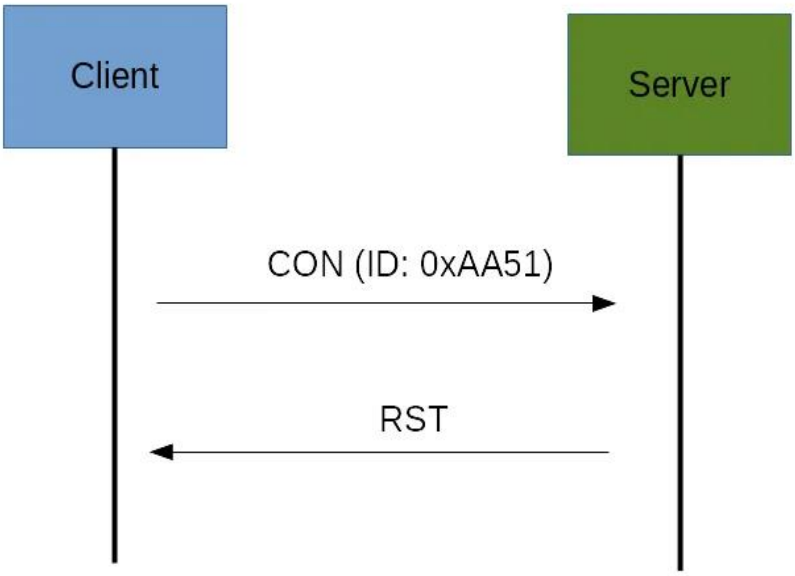
A **confirmable message** is a reliable message. When exchanging messages between two endpoints, these messages can be reliable. In CoAP, a reliable message is obtained using a Confirmable message (CON). Using this kind of message, the client can be sure that the message will arrive at the server. A Confirmable message is sent again and again until the other party sends an acknowledge message (ACK). The ACK message contains the same ID of the confirmable message (CON).

The picture below shows the message exchange process:



coap messages ack

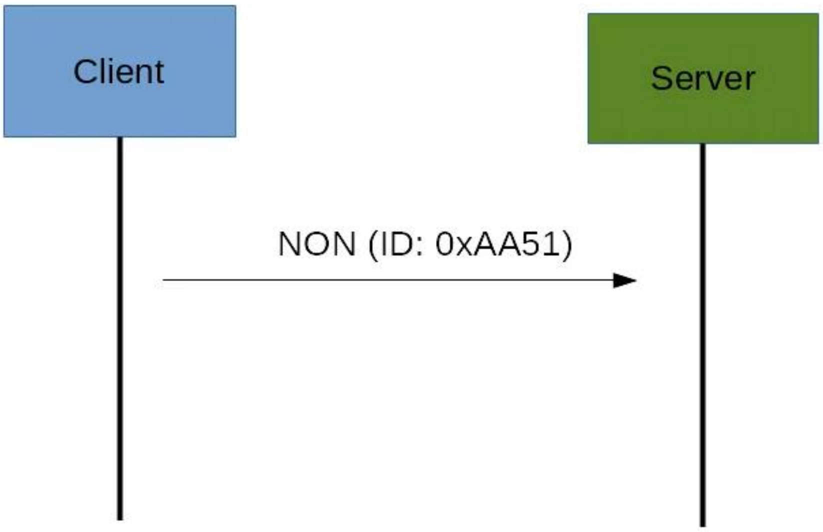
If the server has troubles managing the incoming request, it can send back a Rest message (RST) instead of the Acknowledge message (ACK):



coap messages reset (RST)

The other message category is the Non-confirmable (NON) messages. These are messages that don’t require an Acknowledge by the server. They are unreliable messages or in other words messages that do not contain critical information that must be delivered to the server. To this category belongs messages that contain values read from sensors.

Even if these messages are unreliable, they have a unique ID.

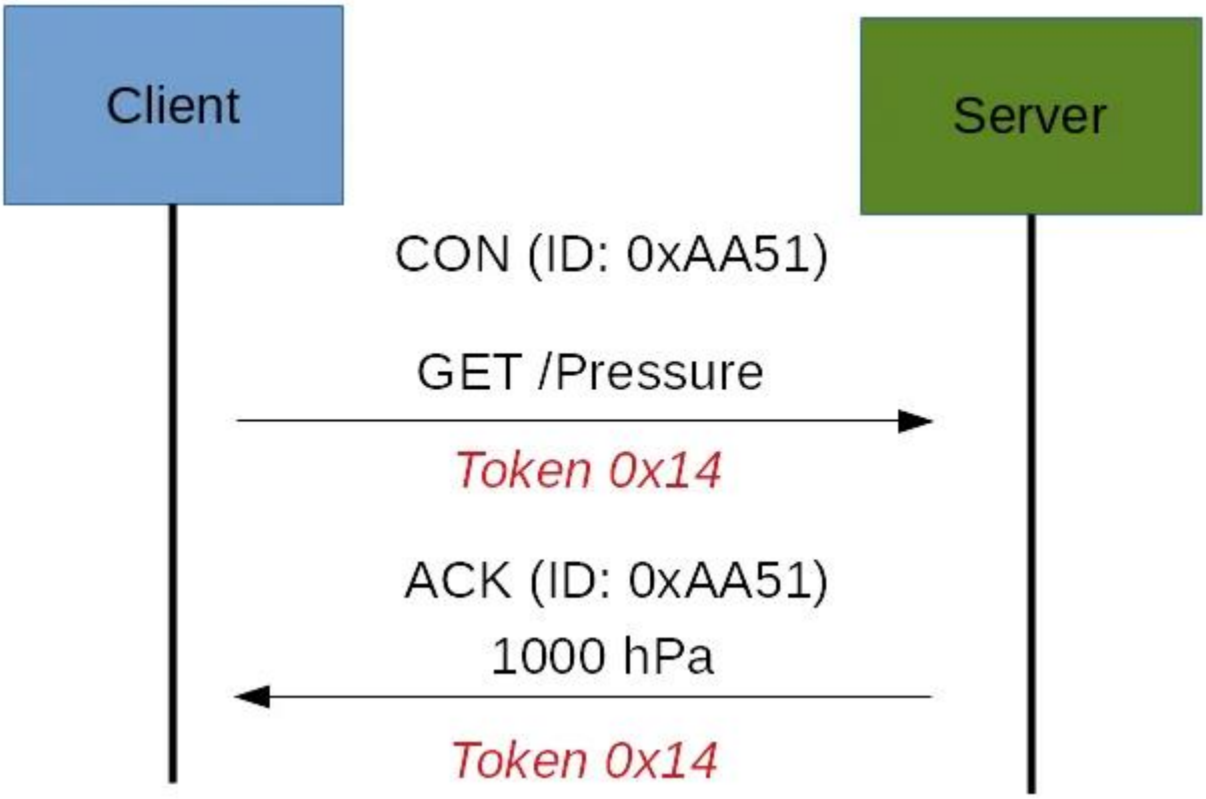


coap messages non

CoAp Request/Response Model

The CoAP Request/Response is the second layer in the CoAP abstraction layer. The request is sent using a Confirmable (CON) or Non-Confirmable (NON) message. There are several scenarios depending on if the server can answer immediately to the client request or the answer if not available.

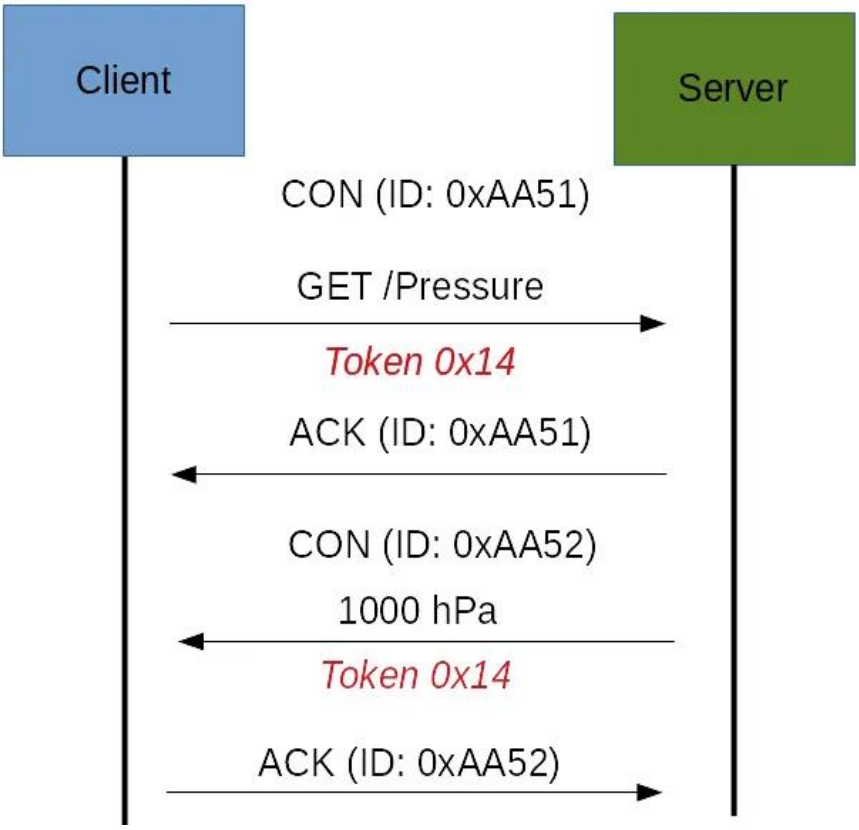
If the server can answer immediately to the client request, then if the request is carried using a **Confirmable message (CON),** the server sends back to the client an Acknowledge message containing the response or the error code:



request ack con

As you can notice in the CoAP message, there is a Token. The Token is different from the Message-ID and it is used to match the request and the response.

If the server can't answer to the request coming from the client immediately, then it sends an Acknowledge message with an empty response. As soon as the response is available, then the server sends a new Confirmable message to the client containing the response. At this point, the client sends back an Acknowledge message:



request ack con async

If the request coming from the client is carried using a NON-confirmable message, then the server answer using a NON-confirmable message.

XMPP - eXtensible Messaging and Presence Protocol

XMPP is a short form for Extensible Messaging Presence Protocol. It’s protocol for streaming XML elements over a network in order to exchange messages and presence information in close to real time. This protocol is mostly used by instant messaging applications like WhatsApp.

Let’s dive into each character of word XMPP:

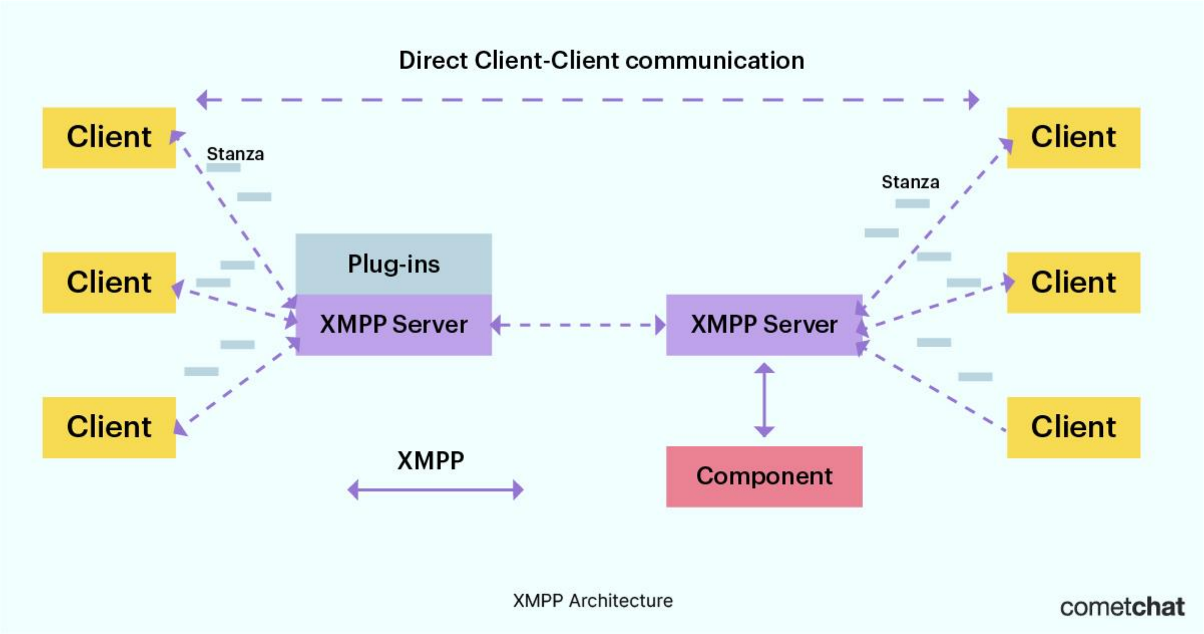
X : It means eXtensible. XMPP is a open source project which can be changed or extended according to the need.

M : XMPP is designed for sending messages in real time. It has very efficient push mechanism compared to other protocols.

P : It determines whether you are online/offline/busy. It indicates the state.

P : XMPP is a protocol, that is, a set of standards that allow systems to communicate with each other.

XMPP Architecture



XMPP works on a client-server architecture. This means that when you send a message via XMPP, it's first sent to a server which then routes it to the correct client (user).

How does it know which client to send your message to?

each client connected to an XMPP server is assigned a unique identifier, known as a Jabber ID. The current format of the unique identifier is user@domain.com/resource.

- The user and domain parts of the unique identifier are pretty straightforward. User is the username of the person and domain.com is the domain of the client that’s sending the message.

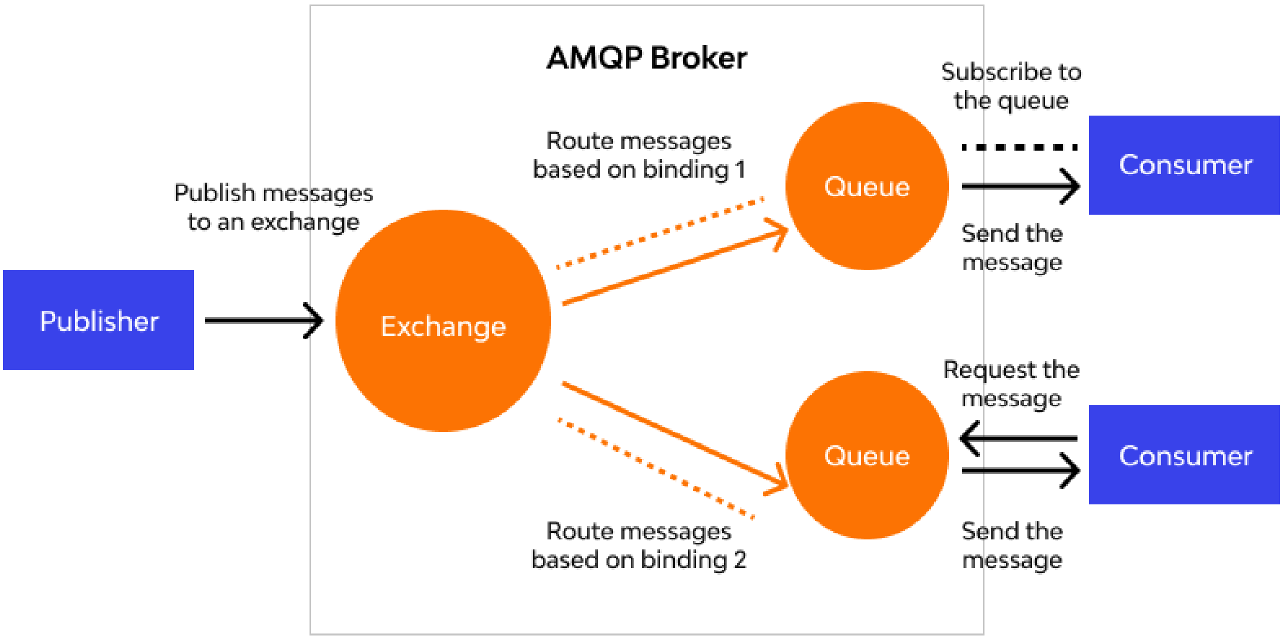
- Resource refers to the type of device on which the message was sent. For example, mobile vs web. This part of the unique identifier is optional and is often only defined in situations where the client doesn't support every device.

Using this unique identifier, an XMPP server can route each message to the correct client. When a client initiates a session with an XMPP server, it opens a persistent TCP connection and starts an XML stream to the server. Once the client is identified by the server (via the unique identifier) and the connection is accepted, the server then opens an additional XML stream—with this stream going back to the client. The end result is a bi-directional stream of XML data.

AMQP - Advanced Message Queueing Protocol

- AMQP is an acronym used for the Advanced Message Queuing Protocol. It is a protocol that is used for communication between applications. It is a lightweight, protocol which supports the applications for transfer of data. This protocol is used for its scalability and modularity with the technologies.

- This advanced message queuing is a suitable protocol for the message-oriented middleware environments. This was developed by John Hara from JP Morgan Chase, London. This IoT communication protocol useful for the exchange of reliable message can be done with this AMQP.
- The publisher can communicate with subscriber through AMQP Broker. The messages from the publisher can be store in the queue of AMQP and as per the message queue and order, they will be forwarded to the relevant consumer with proper security system line. AMQP has the following three capabilities which make it more reliable and secure. This protocol has the below processing chain.



Exchange: Receives the messages from the publishers and based on priorities they are forwarded to queues.

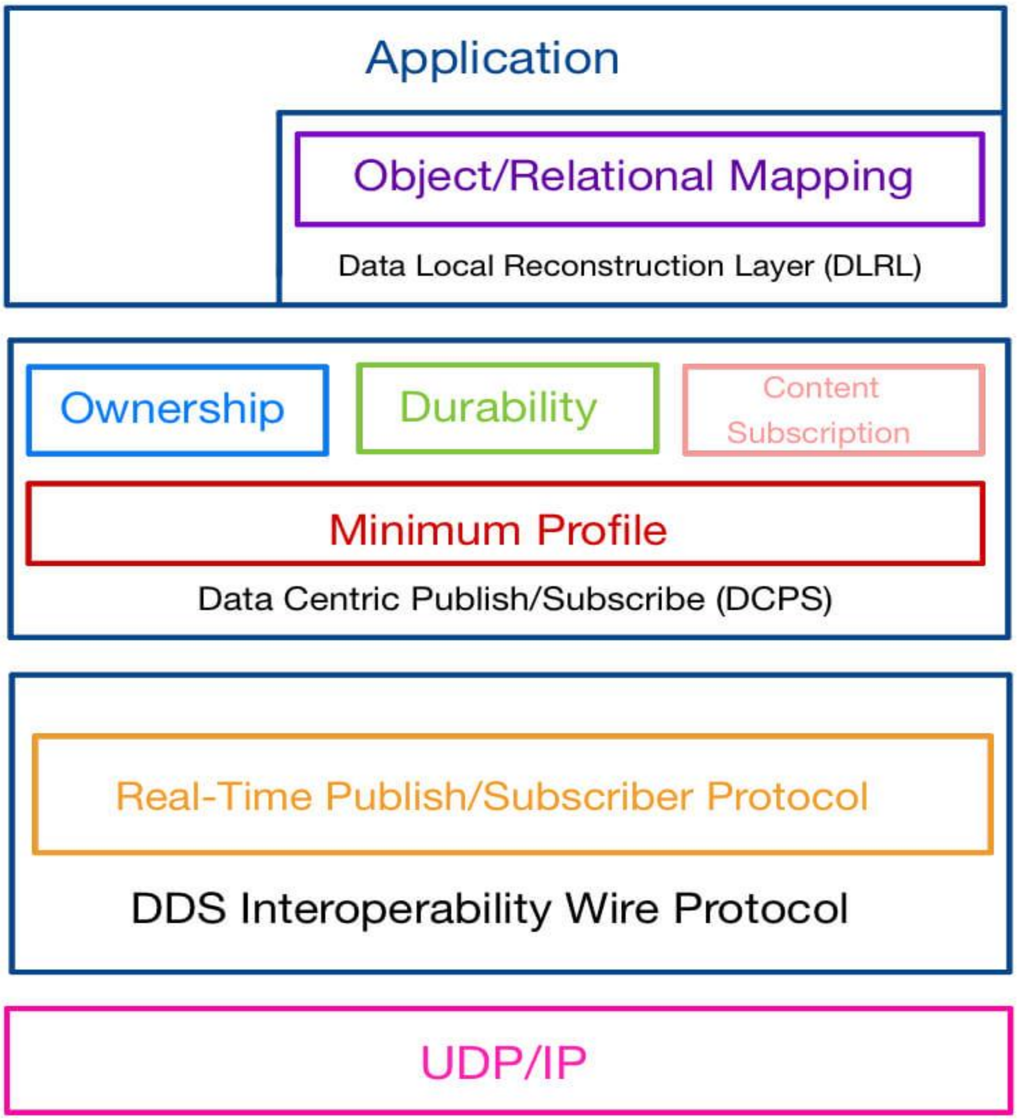
Queue: Stores the messages until they are properly processed with client software.

Binding: The connection between the exchange and queue will state by this binding component.

Data Distribution Service (DDS):

DDS is another publish-subscribe protocol, but it is different from MQTT which connects them to server but here DDS protocol is a Broker-less architecture that's why it is a high speed and high-performance protocol than MQTT as it is not dependent upon any intermediary system.

DDS PROTOCOL

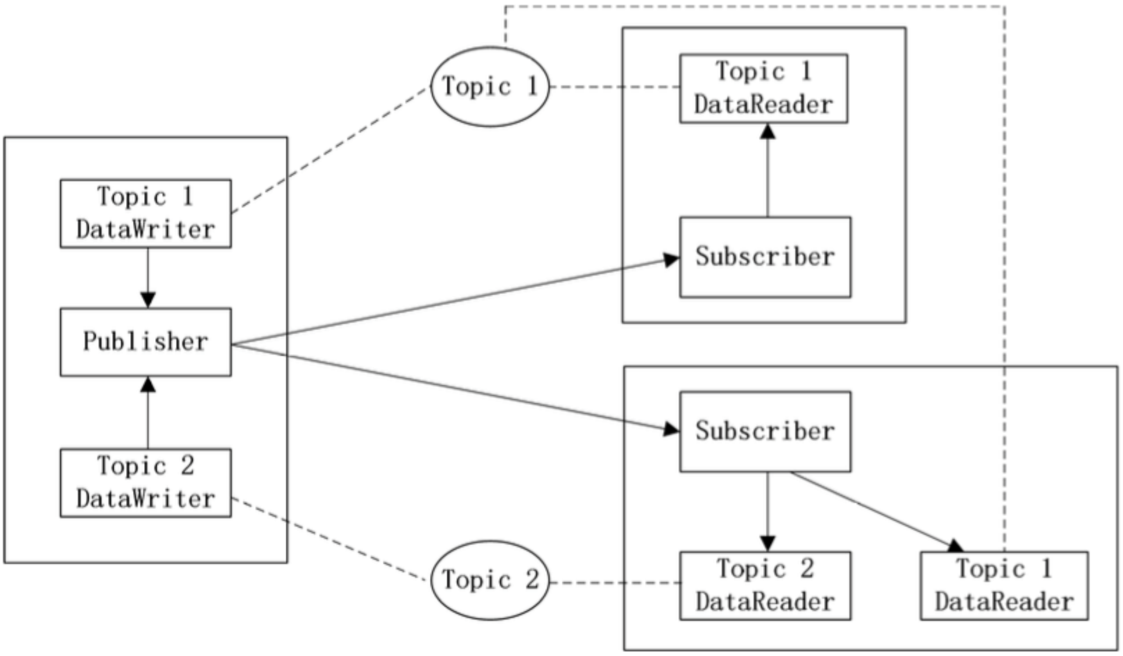

Application
Object/Relational Mapping
Data Local Reconstruction Layer (DLRL)
Ownership
Durability
Content Subscription
Minimum Profile
Data Centric Publish/Subscribe (DCPS)
Real-Time Publish/Subscriber Protocol
DDS Interoperability Wire Protocol
UDP/IP

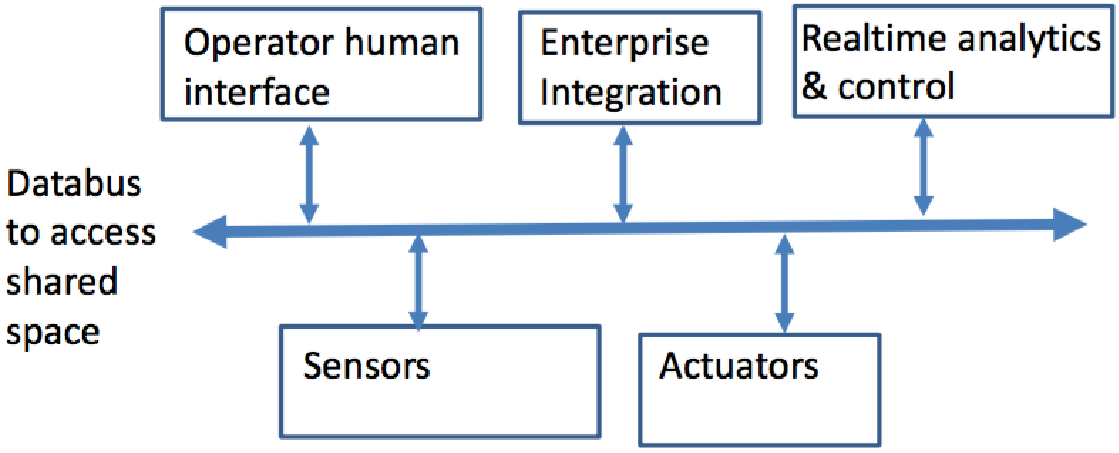
DDS Protocol Stack

- The protocol uses broker-less architecture in the Internet of Things, unlike MQTT and CoAP protocols.
- It is an IoT protocol developed by Object Management Group for Machine to Machine Communication.
- It uses a publish-subscribe methodology for exchanging data.
- It uses multicasting to bring high-quality QoS to the applications.

- It is designed by the OMG (Object Management Group) for device to device communications. This protocol has two fundamental sublayers i.e., Data-Centric Publish-Subscribe (DCPS) and Data Local Reconstruction Layer (DLRL).
- The DCPS stands for Data-Centric Publish-Subscribe, it is a standard API for a data-centric, topic-based, and real-time publish/subscribe layer. The function of this layer is to deliver information to subscribers.
- The DLRL stands for Data Local Reconstruction Layer, it is a standard API for creating object views from the collection of topics. Its function is to provide an interface to DCPS functionalities. This enables sharing of distributed data among devices that are IoT enables.

The architecture DDS is shown below



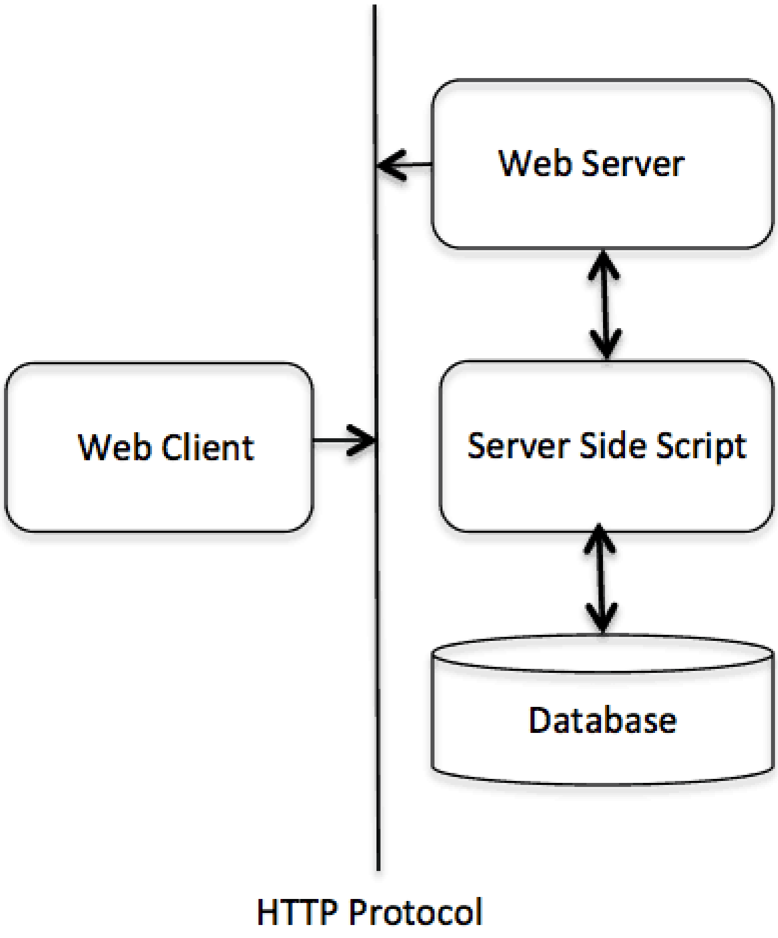


HTTP

- HTTP (Hyper Text Transfer Protocol) is an application-layer protocol used for communicating between a client and a server.
- HTTP is a request/response protocol. It specifies what clients can send to a server, and what they can expect to receive back
- Originally HTTP was intended to transfer HTML documents from servers to browsers, but it's now used for many different kinds of media.

Basic Architecture

The following diagram shows a very basic architecture of a web application and depicts where HTTP sits:



The HTTP protocol is a request/response protocol based on the client/server based architecture where web browsers, robots and search engines, etc. act like HTTP clients, and the Web server acts as a server.

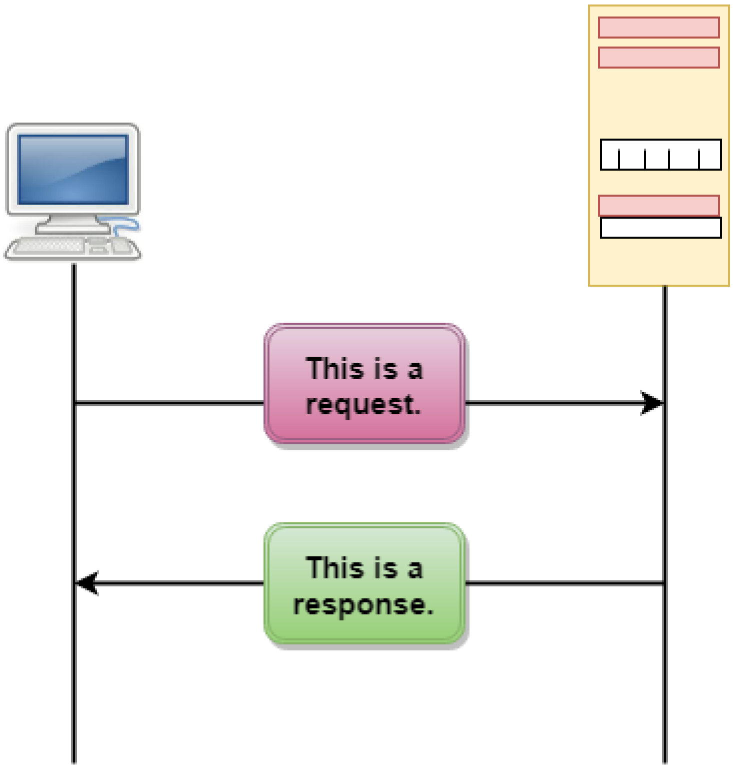
Client

The HTTP client sends a request to the server in the form of a request method, URI, and protocol version, followed by a MIME-like message containing request modifiers, client information, and possible body content over a TCP/IP connection.

Server

The HTTP server responds with a status line, including the message's protocol version and a success or error code, followed by a MIME-like message containing server information, entity meta information, and possible entity-body content.

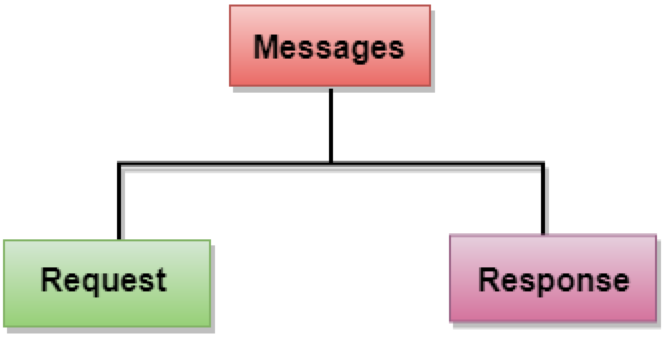
HTTP Transactions



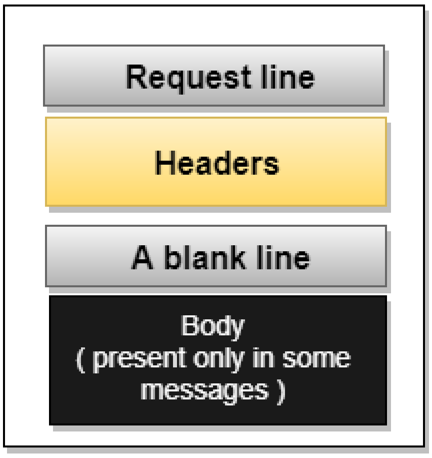
The above figure shows the HTTP transaction between client and server. The client initiates a transaction by sending a request message to the server. The server replies to the request message by sending a response message.

Messages

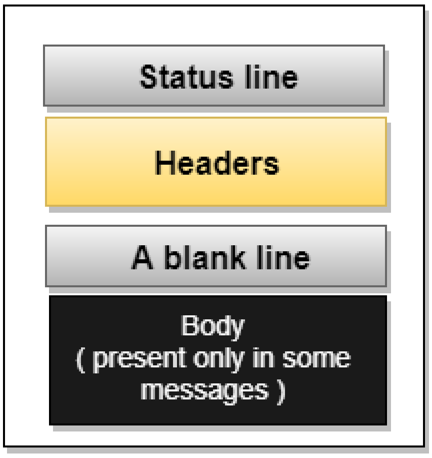
HTTP messages are of two types: request and response. Both the message types follow the same message format.



Request Message: The request message is sent by the client that consists of a request line, headers, and sometimes a body.

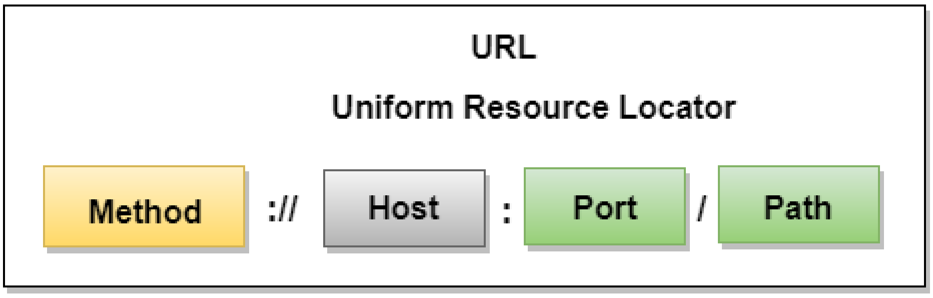


Response Message: The response message is sent by the server to the client that consists of a status line, headers, and sometimes a body.



Uniform Resource Locator (URL)

- A client that wants to access the document in an internet needs an address and to facilitate the access of documents, the HTTP uses the concept of Uniform Resource Locator (URL).
- The Uniform Resource Locator (URL) is a standard way of specifying any kind of information on the internet.
- The URL defines four parts: method, host computer, port, and path.



- **Method:** The method is the protocol used to retrieve the document from a server. For example, HTTP.
- **Host:** The host is the computer where the information is stored, and the computer is given an alias name. Web pages are mainly stored in the computers and the computers are given an alias name that begins with the characters "www". This field is not mandatory.
- **Port:** The URL can also contain the port number of the server, but it's an optional field. If the port number is included, then it must come between the host and path and it should be separated from the host by a colon.
- **Path:** Path is the pathname of the file where the information is stored. The path itself contain slashes that separate the directories from the subdirectories and files.

What is IP?

An IP stands for internet protocol. An IP address is assigned to each device connected to a network. Each device uses an IP address for communication. It also behaves as an identifier as this address is used to identify the device on a network. It defines the technical format of the packets. Mainly, both the networks, i.e., IP and TCP, are combined together, so together, they are referred to as a TCP/IP. It creates a virtual connection between the source and the destination.

We can also define an IP address as a numeric address assigned to each device on a network. An IP address is assigned to each device so that the device on a network can be identified uniquely. To facilitate the routing of packets, TCP/IP protocol uses a 32-bit logical address known as IPv4(Internet Protocol version 4).

An IP address consists of two parts, i.e., the first one is a network address, and the other one is a host address.

There are two types of IP addresses:

- IPv4

- IPv6

What is IPv4?

IPv4 is a version 4 of IP. It is a current version and the most commonly used IP address. It is a 32-bit address written in four numbers separated by 'dot', i.e., periods. This address is unique for each device.

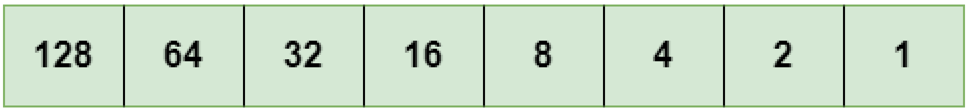
For example, **66.94.29.13**

The above example represents the IP address in which each group of numbers separated by periods is called an Octet. Each number in an octet is in the range from 0-255. This address can produce 4,294,967,296 possible unique addresses.

In today's computer network world, computers do not understand the IP addresses in the standard numeric format as the computers understand the numbers in binary form only. The binary number can be either 1 or 0. The IPv4 consists of four sets, and these sets represent the octet. The bits in each octet represent a number.

Each bit in an octet can be either 1 or 0. If the bit the 1, then the number it represents will count, and if the bit is 0, then the number it represents does not count.

Representation of 8 Bit Octet

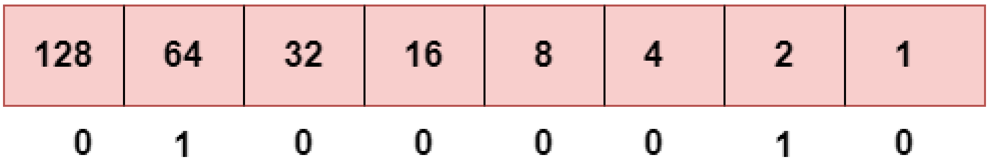


128	64	32	16	8	4	2	1

The above representation shows the structure of 8- bit octet.

Now, we will see how to obtain the binary representation of the above IP address, i.e., 66.94.29.13

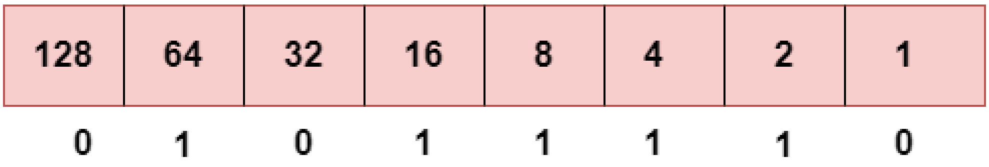
Step 1: First, we find the binary number of 66.



128	64	32	16	8	4	2	1
0	1	0	0	0	0	1	0

To obtain 66, we put 1 under 64 and 2 as the sum of 64 and 2 is equal to 66 (64+2=66), and the remaining bits will be zero, as shown above. Therefore, the binary bit version of 66 is 01000010.

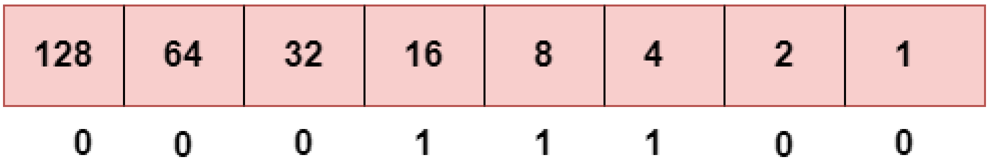
Step 2: Now, we calculate the binary number of 94.



128	64	32	16	8	4	2	1
0	1	0	1	1	1	1	0

To obtain 94, we put 1 under 64, 16, 8, 4, and 2 as the sum of these numbers is equal to 94, and the remaining bits will be zero. Therefore, the binary bit version of 94 is 01011110.

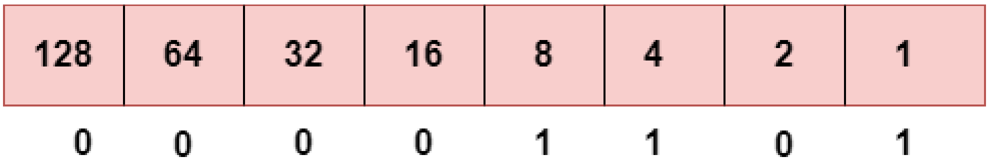
Step 3: The next number is 29.



128	64	32	16	8	4	2	1
0	0	0	1	1	1	0	0

To obtain 29, we put 1 under 16, 8, 4, and 1 as the sum of these numbers is equal to 29, and the remaining bits will be zero. Therefore, the binary bit version of 29 is 00011101.

Step 4: The last number is 13.



128	64	32	16	8	4	2	1
0	0	0	0	1	1	0	1

To obtain 13, we put 1 under 8, 4, and 1 as the sum of these numbers is equal to 13, and the remaining bits will be zero. Therefore, the binary bit version of 13 is 00001101.

Drawback of IPv4

Currently, the population of the world is 7.6 billion. Every user is having more than one device connected with the internet, and private companies also rely on the internet. As we know that IPv4 produces 4 billion addresses, which are not enough for each device connected to the internet on a planet. Although the various techniques were invented, such as variable- length mask, network address translation, port address translation, classes, inter-domain translation, to conserve the bandwidth of IP address and slow down the depletion of an IP address. In these techniques, public IP is converted into a private IP due to which the user having public IP can also use the internet. But still, this was not so efficient, so it gave rise to the development of the next generation of IP addresses, i.e., IPv6.

What is IPv6?

IPv4 produces 4 billion addresses, and the developers think that these addresses are enough, but they were wrong. IPv6 is the next generation of IP addresses. The main difference between IPv4 and IPv6 is the address size of IP addresses. The IPv4 is a 32-bit address, whereas IPv6 is a 128-bit hexadecimal address. IPv6 provides a large address space, and it contains a simple header as compared to IPv4.

It provides transition strategies that convert IPv4 into IPv6, and these strategies are as follows:

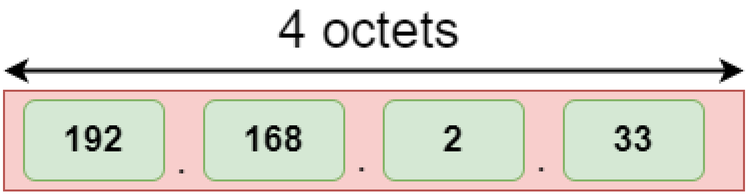
- **Dual stacking:** It allows us to have both the versions, i.e., IPv4 and IPv6, on the same device.
- **Tunneling:** In this approach, all the users have IPv6 communicates with an IPv4 network to reach IPv6.
- **Network Address Translation:** The translation allows the communication between the hosts having a different version of IP.

This hexadecimal address contains both numbers and alphabets. Due to the usage of both the numbers and alphabets, IPv6 is capable of producing over 340 undecillion ($3.4*10^{38}$) addresses.

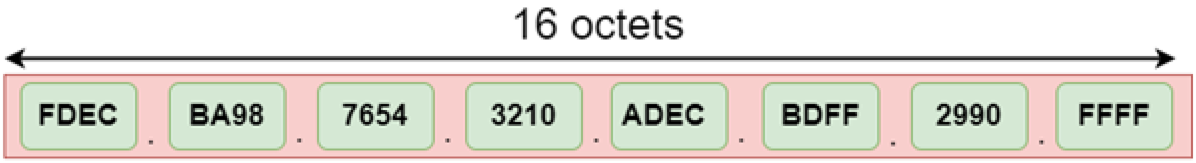
IPv6 is a 128-bit hexadecimal address made up of 8 sets of 16 bits each, and these 8 sets are separated by a colon. In IPv6, each hexadecimal character represents 4 bits. So, we need to convert 4 bits to a hexadecimal number at a time

Address format

The address format of IPv4:



The address format of IPv6:



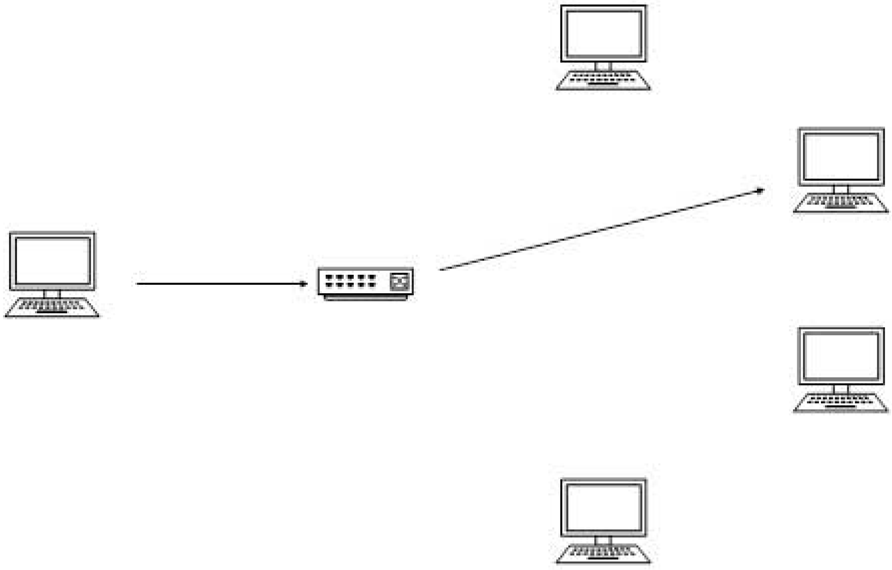
The above diagram shows the address format of IPv4 and IPv6. An IPv4 is a 32-bit decimal address. It contains 4 octets or fields separated by 'dot', and each field is 8-bit in size. The number that each field contains should be in the range of 0-255. Whereas an IPv6 is a 128-bit hexadecimal address. It contains 8 fields separated by a colon, and each field is 16-bit in size.

IPv6 - Addressing Modes

addressing mode refers to the mechanism of hosting an address on the network. IPv6 offers several types of modes by which a single host can be addressed. More than one host can be addressed at once or the host at the closest distance can be addressed.

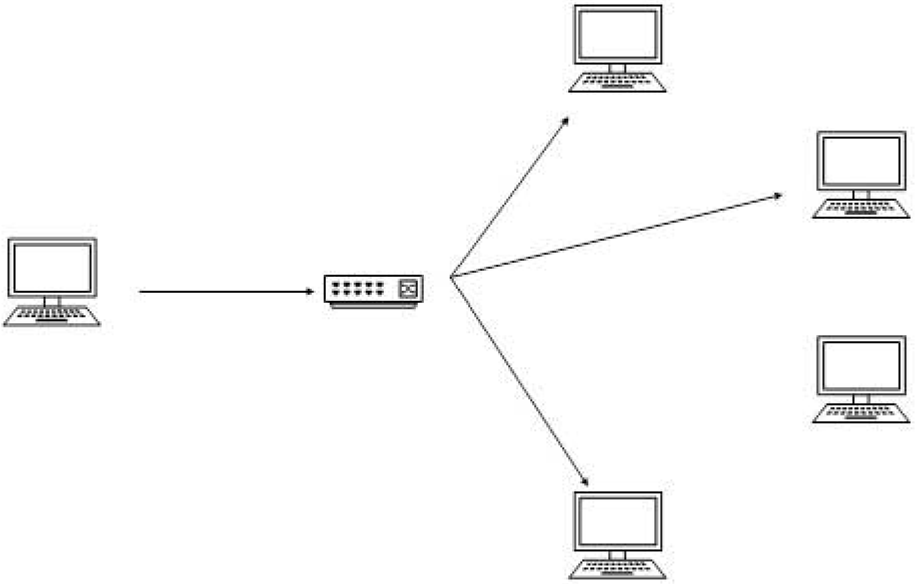
Unicast

In unicast mode of addressing, an IPv6 interface (host) is uniquely identified in a network segment. The IPv6 packet contains both source and destination IP addresses. A host interface is equipped with an IP address which is unique in that network segment.When a network switch or a router receives a unicast IP packet, destined to a single host, it sends out one of its outgoing interface which connects to that particular host.



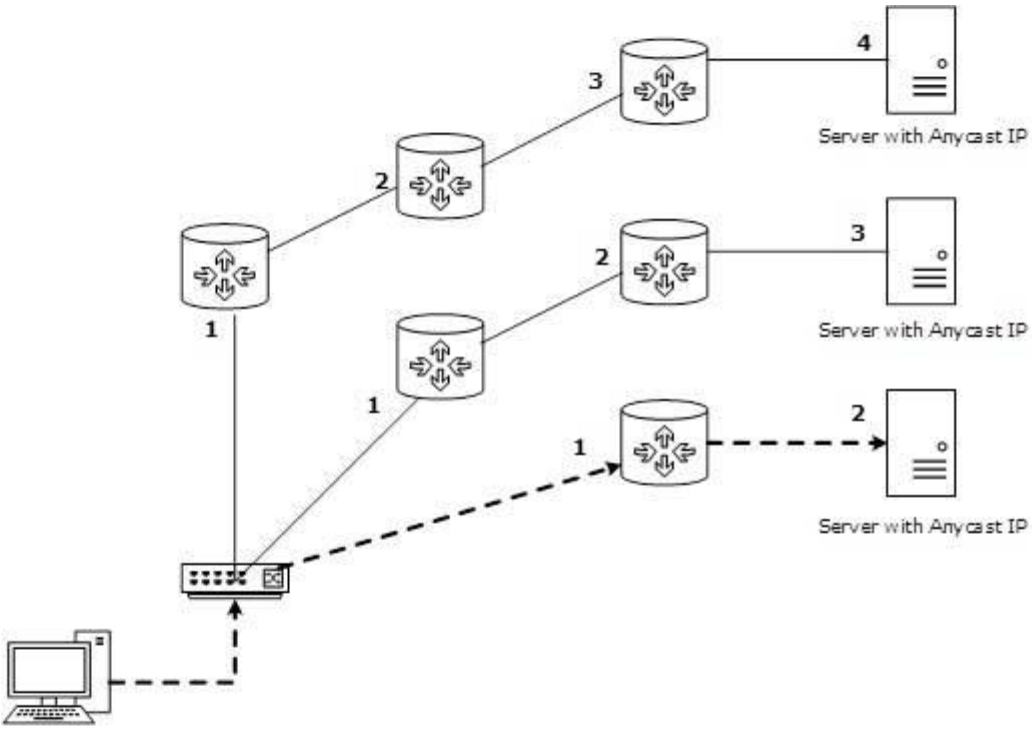
Multicast

The IPv6 multicast mode is same as that of IPv4. The packet destined to multiple hosts is sent on a special multicast address. All the hosts interested in that multicast information, need to join that multicast group first. All the interfaces that joined the group receive the multicast packet and process it, while other hosts not interested in multicast packets ignore the multicast information.



Anycast

IPv6 has introduced a new type of addressing, which is called Anycast addressing. In this addressing mode, multiple interfaces (hosts) are assigned same Anycast IP address. When a host wishes to communicate with a host equipped with an Anycast IP address, it sends a Unicast message. With the help of complex routing mechanism, that Unicast message is delivered to the host closest to the Sender in terms of Routing cost.



Let’s take an example of TutorialPoints.com Web Servers, located in all continents. Assume that all the Web Servers are assigned a single IPv6 Anycast IP Address.

Now when a user from Europe wants to reach TutorialsPoint.com the DNS points to the server that is physically located in Europe itself. If a user from India tries to reach Tutorialspoint.com, the DNS will then point to the Web Server physically located in Asia. Nearest or Closest terms are used in terms of Routing Cost.

In the above picture, when a client computer tries to reach a server, the request is forwarded to the server with the lowest Routing Cost.

Address Structure

An IPv6 address is made of 128 bits divided into eight 16-bits blocks. Each block is then converted into 4-digit Hexadecimal numbers separated by colon symbol.

For example, the below is 128 bit IPv6 address represented in binary format and divided into eight 16-bits blocks:

```
0010000000000001 0000000000000000 0011001000110100 1101111111100001
0000000001100011 0000000000000000 0000000000000000 1111111011111011
```

Each block is then converted into Hexadecimal and separated by ':' symbol:

```
2001:0000:3238:DFE1:0063:0000:0000:FEFB
```

Even after converting into Hexadecimal format, IPv6 address remains long. IPv6 provides some rules to shorten the address. These rules are:

Rule:1 Discard leading Zero(es):

In Block 5, 0063, the leading two 0s can be omitted, such as (5th block):

```
2001:0000:3238:DFE1:63:0000:0000:FEFB
```

Rule:2 If two of more blocks contains consecutive zeroes, omit them all and replace with double colon sign ::, such as (6th and 7th block):

```
2001:0000:3238:DFE1:63::FEFB
```

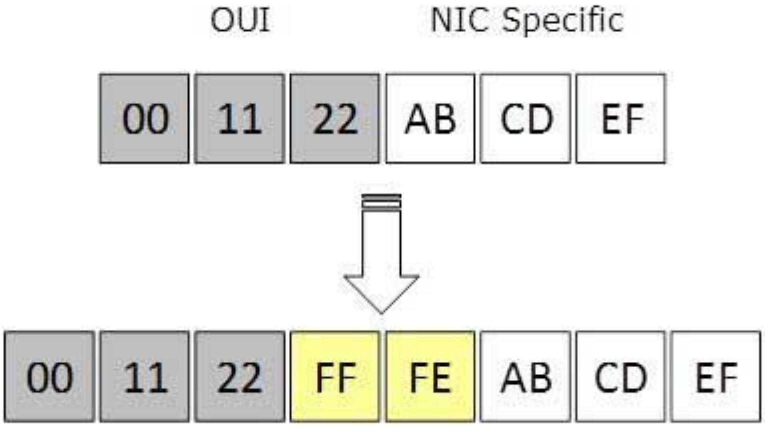
Consecutive blocks of zeroes can be replaced only once by :: so if there are still blocks of zeroes in the address they can be shrink down to single zero, such as (2nd block):

```
2001:0:3238:DFE1:63::FEFB
```

Interface ID

IPv6 has three different type of Unicast Address scheme. The second half of the address (last 64 bits) is always used for Interface ID. MAC address of a system is composed of 48-bits and represented in Hexadecimal. MAC address is considered to be uniquely assigned worldwide. Interface ID takes advantage of this uniqueness of MAC addresses. A host can auto-configure its Interface ID by using IEEE's Extended

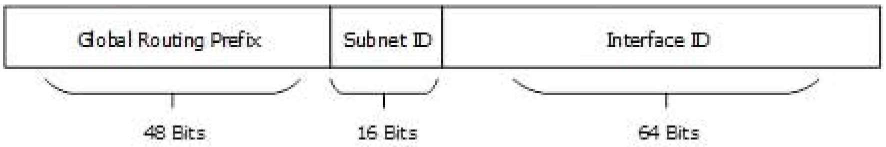
Unique Identifier (EUI-64) format. First, a Host divides its own MAC address into two 24-bits halves. Then 16-bit Hex value 0xFFFE is sandwiched into those two halves of MAC address, resulting in 64-bit Interface ID.



[*Image: EUI-64 Interface ID*]

Global Unicast Address

This address type is equivalent to IPv4's public address. Global Unicast addresses in IPv6 are globally identifiable and uniquely addressable.

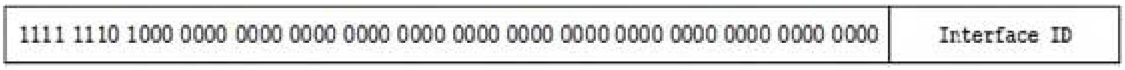


[*Image: Global Unicast Address*]

Global Routing Prefix: The most significant 48-bits are designated as Global Routing Prefix which is assigned to specific Autonomous System. Three most significant bits of Global Routing Prefix is always set to 001.

Link-Local Address

Auto-configured IPv6 address is known as Link-Local address. This address always starts with FE80. First 16 bits of Link-Local address is always set to 1111 1110 1000 0000 (FE80). Next 48-bits are set to 0, thus:



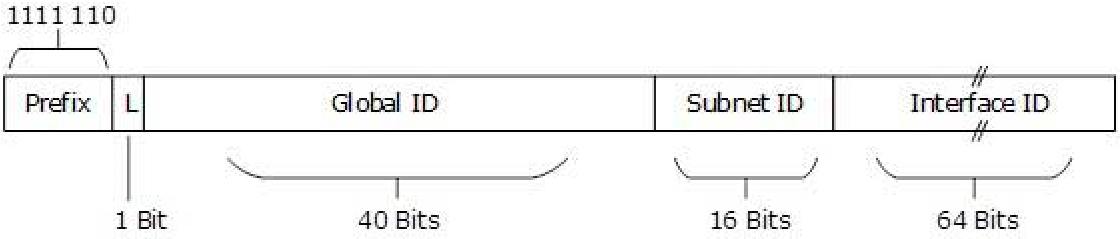
1111 1110 1000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000	Interface ID

[*Image: Link-Local Address*]

Link-Local addresses are used for communication among IPv6 hosts on a link (broadcast segment) only. These addresses are not routable so a Router never forwards these addresses outside the link.

Unique-Local Address

This type of IPv6 address which is though globally unique, but it should be used in local communication. This address has second half of Interface ID and first half is divided among Prefix, Local Bit, Global ID and Subnet ID.



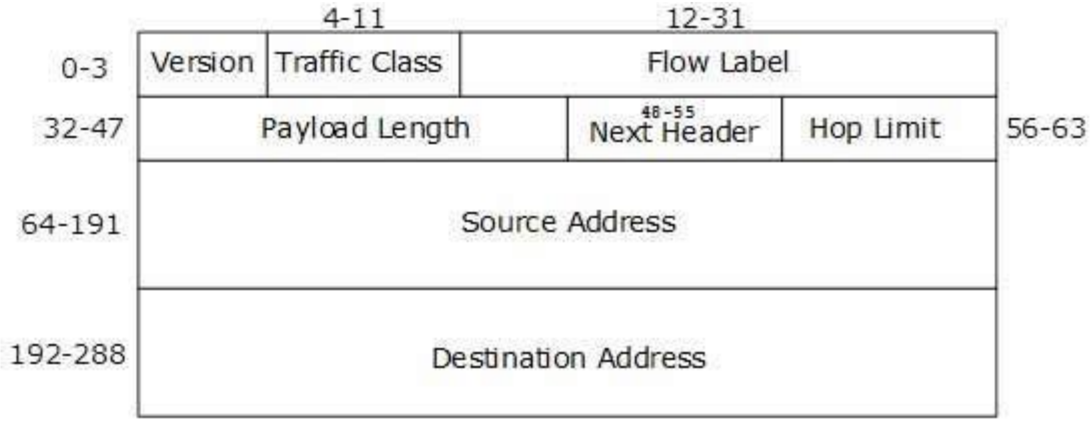
[*Image: Unique-Local Address*]

Prefix is always set to 1111 110. L bit, which is set to 1 if the address is locally assigned. So far the meaning of L bit to 0 is not defined. Therefore, Unique Local IPv6 address always starts with 'FD'.

Headers

The wonder of IPv6 lies in its header. IPv6 address is 4 times larger than IPv4 but the IPv6 header is only 2 times larger than that of IPv4. IPv6 headers have one Fixed Header and zero or more Optional (Extension) Headers. All necessary information which is essential for a router is kept in Fixed Header. Extension Header contains optional information which helps routers to understand how to handle a packet/flow.

Fixed Header



[*Image: IPv6 Fixed Header*]

IPv6 fixed header is 40 bytes long and contains the following information.

S.N.	Field & Description
1	**Version** (4-bits): This represents the version of Internet Protocol, i.e. 0110.
2	**Traffic Class** (8-bits): These 8 bits are divided into two parts. Most significant 6 bits are used for Type of Service, which tells the Router what services should be provided to this packet. Least significant 2 bits are used for Explicit Congestion Notification (ECN).
3	**Flow Label** (20-bits): This label is used to maintain the sequential flow of the packets belonging to a communication. The source labels the sequence which helps the router to identify that this packet belongs to a specific flow of information. This field helps to avoid re-ordering of data packets. It is designed for streaming/real-time media.
4	**Payload Length** (16-bits): This field is used to tell the routers how much information this packet contains in its payload. Payload is composed of Extension Headers and Upper Layer data. With 16 bits, up to 65535 bytes can be indicated but if Extension Headers contain Hop-by-Hop Extension Header than payload may exceed 65535 bytes and this field is set to 0.
5	**Next Header** (8-bits): This field is used to indicate either the type of Extension Header, or if Extension Header is not present then it indicates the Upper Layer PDU. The values for the type of Upper Layer PDU is same as IPv4's.
6	**Hop Limit** (8-bits): This field is used to stop packet to loop in the network infinitely. This is same as TTL in IPv4. The value of Hop Limit field is decremented by 1 as it passes a link (router/hop). When the field reaches 0 the packet is discarded.
7	**Source Address** (128-bits): This field indicates the address of originator of the packet.
8	**Destination Address** (128-bits): This field provides the address of intended recipient of the packet.

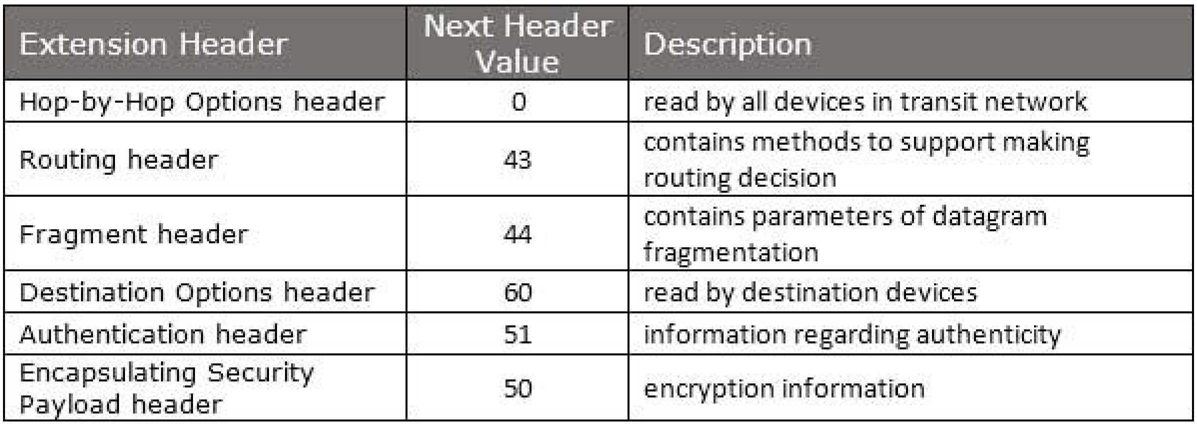
Extension Headers

In IPv6, the Fixed Header contains only information which is necessary and avoiding information which is either not required or is rarely used. All such information, is put between the Fixed Header and Upper layer header in the form of Extension Headers. Each Extension Header is identified by a distinct value.

When Extension Headers are used, IPv6 Fixed Header's Next Header field points to the first Extension Header. If there is one more Extension Header, then first Extension Header's 'Next-Header' field point to the second one, and so on. The last Extension Header's 'Next-Header' field point to Upper Layer Header. Thus all headers from point to the next one in a linked list manner.

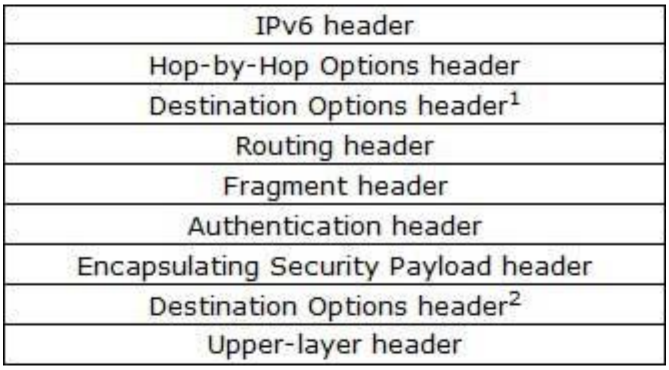
If the Next Header field contains value 59, it indicates that there's no header after this header, not even Upper Layer Header.

The following Extension Headers must be supported as per RFC 2460:



Extension Header	Next Header Value	Description
Hop-by-Hop Options header	0	read by all devices in transit network
Routing header	43	contains methods to support making routing decision
Fragment header	44	contains parameters of datagram fragmentation
Destination Options header	60	read by destination devices
Authentication header	51	information regarding authenticity
Encapsulating Security Payload header	50	encryption information

The sequence of Extension Headers should be:

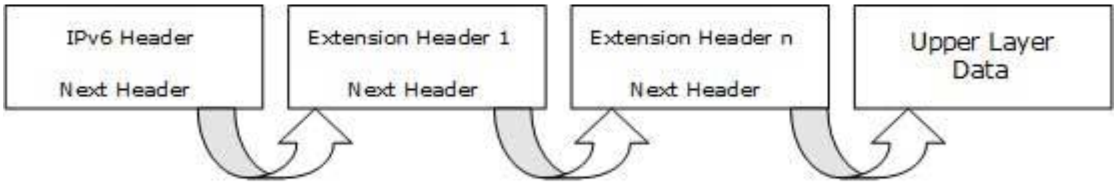


IPv6 header
Hop-by-Hop Options header
Destination Options header[1]
Routing header
Fragment header
Authentication header
Encapsulating Security Payload header
Destination Options header[2]
Upper-layer header

These headers:

- 1. Should be processed by First and subsequent destinations.
- 2. Should be processed by Final Destination.

Extension Headers are arranged one after another in a Linked list manner, as depicted in the diagram below:



[*Image: Extension Headers Connected Format*]

6LoWPAN

The 6LoWPAN protocol refers to IPv6 Low Power Personal Area Network which uses a lightweight IP-based communication to travel over low data rate networks. It has limited processing ability to transfer information wirelessly using an internet protocol. So, it is mainly used for home and building automation. The 6LoWPAN protocol operates only within the 2.4 GHz frequency range with 250 kbps transfer rate. It has a maximum length of 128-bit header packets.

IPv6 Low Power Wireless Personal Area Network (6LoWPAN) is an IPv6 standard based network layer protocol for Wireless Personal Area Networks. Based on 802.15.4 protocol at physical layer, the standard has been developed for addressing of IOT sensors and devices in a Wireless Sensor Network (WSN). This protocol is a modified version of IPv6 with intention to implement Internet protocol to each and every devices (constrained devices as well as large devices) and the low power devices with limited capabilities like less memory, lossy network etc. IPv6 operates only at 2.4 GHz frequency range with 250 Kbps transfer data rate.

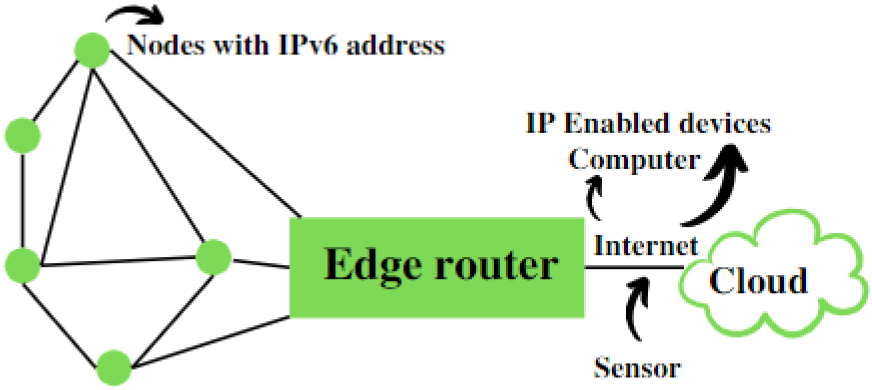
6LoWPAN networks connect to the Internet via a gateway (WiFi or Ethernet), which does some process for protocol conversion so that device can communicate with Internet. Specifically, the adaptation layer performs the following three optimizations in order to reduce communication overhead –

1) Header Compression — Ipv6 supports packet header length of 127 byte. So, 6loWPAN defines header compression of IPv6 packets for decreasing the overhead of IPv6.

2) Fragmentation — The minimum MTU size (maximum transmission unit) of IPv6 is 1280 bytes. On the other hand, the maximum size of a frame in IEEE 802.15.4 is

127 bytes. Therefore, the IPv6 packet need to be fragmented. This is done by the adaptation layer.

3) Link Layer Forwarding — 6LoWPAN also supports mesh under routing, which is done at the link layer using link level short addresses. This feature can be used to communicate within a 6LoWPAN network.



6LowPAN Security Measure

Security is a major issue for 6LowPAN communication Protocol. There are several attacks issues at the security level of 6LoWPAN which aim is to direct destruction of the network. Since it is the combination of two systems, so, there is a possibility of attack from two sides that targets all the layer of the 6LoWPAN stack (Physical layer, Data link layer, Adaptation layer, Network layer, Transport layer, Application layer).

Properties of 6LowPAN protocol

Standard: RFC6282
Frequency: Used over a variety of other networking media including Bluetooth Smart (2.4GHz) or ZigBee or low-power RF (sub-1GHz)
Range: NA
Data Rates: NA

IoT WIRED COMMUNICATION

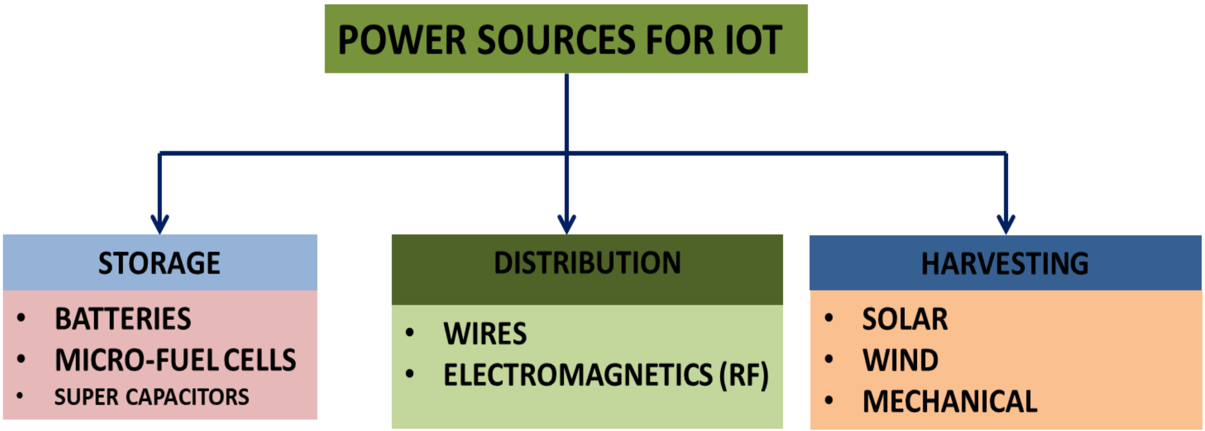
- IoT technology is deployed in many ways so no single network solution is right. It depends on the situation and where the devices are located. Some of the factors affecting the selection of the type of network are
  - network range
  - network bandwidth
  - power usage
  - Interoperability

  - intermittent connectivity
  - security.
- A wired network uses Ethernet cable to connect to the network. The Ethernet cable is in turn connected to a DSL or cable to the network gateway. The wired networks are mature technology and it is easy to get plugged into if you already have phone lines, power lines, and coaxial cable lines.

BENEFITS OF THE WIRED MONITORING DEVICES

  - **Reliability:** Less prone to dropped connections.
  - **Speed:** Wired connectivity is much faster than wireless. Wired data transmissions are not sensitive to distances and has no effect on the performance of the connection.
  - **Security:** Cannot be hacked or tapped
- **DISADVANTAGES**
  - **Cost:** Wired connections are more expensive than wireless due to the cost of the wire, labor costs for installation.
  - **Mobility:** No mobility
  - **Scalability:** Scalability would be an issue not only for networks to be up and running quickly but also for the planning and cost purposes.

IoT POWER SOURCES


POWER SOURCES FOR IOT
STORAGE
• BATTERIES
• MICRO-FUEL CELLS
• SUPER CAPACITORS
DISTRIBUTION
• WIRES
• ELECTROMAGNETICS (RF)
HARVESTING
• SOLAR
• WIND
• MECHANICAL